Navegando no universo da programação

Conselhos para aproveitar e contribuir para a área tech

Casa do Código | alura

WILLIAM OLIVEIRA

Sumário

- ISBN
- Agradecimentos
- Sobre o autor
- Introdução
- 1. Não seja babaca de comunidade
- 2. Por que você desistiu de ser cientista?
- 3. Assumindo o papel de protagonista em nossa aprendizagem
- 4. Problemas ou armadilhas de sermos autodidatas
- 5. Pare de procurar conteúdo e comece a praticar!
- 6. Sobre os requisitos das vagas de emprego em programação
- 7. Escolha um trabalho que você ama e nunca terá que trabalhar. Mentira!
- 8. Quando abandonar um emprego
- 9. Devolva para as comunidades
- 10. Por que existem cursos de programação exclusivos para LGBTs e outras minorias?
- 11. A vaidade em engenharia de software
- 12. Preconceito com tecnologias
- 13. Conclusão

ISBN

Impresso: 978-85-5519-308-8

Digital: 978-85-5519-307-1

Caso você deseje submeter alguma errata ou sugestão, acesse http://erratas.casadocodigo.com.br.

Agradecimentos

Este livro nasceu da minha contribuição com as comunidades de programação.

Então, aqui eu deixo registrado meus agradecimentos a todas as pessoas que escrevem artigos, prestam mentoria, criam conteúdo no geral e compartilham informação para pessoas que não conseguiriam pagar por isso.

Obrigado, comunidade de programação.

Em um agradecimento mais que especial, gostaria de comentar sobre Carlos Zambrana, pessoa que me incentivou a investir meu tempo em escrever livros para pessoas iniciantes. Graças a Carlos, hoje eu continuo escrevendo e vem muito mais por aí.

Sobre o autor

Olá!

Eu sou William Oliveira, desenvolvedor de software desde 2014, palestrante, escritor, podcaster, patinador e tibiano nas horas vagas.

Nascido na periferia de São Bernardo do Campo, no grande ABC Paulista, encontrei o caminho da programação através das comunidades de tecnologia por volta de 2007. Iniciei no universo da programação como desenvolvedor full-stack, atuando com a linguagem PHP e migrando para front-end quando JavaScript ganhou grandes responsabilidades em aplicações de larga escala, dando à pessoa desenvolvedora de software a possibilidade de pensar em interfaces Web como aplicações, não mais como páginas estáticas.

Apaixonado por open source, software livre, conhecimento aberto, comunidades e afins, fundei o FEMUG-ABC, um grupo de meetups de desenvolvedores front-end que depois deu origem ao maior evento de tecnologia do grande ABC, o ABCDev. Fundei o Training Center, uma comunidade inclusiva focada em ajudar pessoas a entrarem na área de desenvolvimento de software, posteriormente utilizei todo o conhecimento adquirido para investir no *perifaCode*, uma comunidade de programação para pessoas periféricas. Sou voluntário no núcleo de tecnologia do MTST, onde ensinamos e criamos tecnologia para o movimento que mais ajuda pessoas em situação de vulnerabilidade social do país.

Como uma pessoa vinda da favela brasileira, entendo bem a dificuldade das pessoas menos privilegiadas, os grupos sub-representados, e busco apoiar a diversidade e inclusão de minorias na área de programação de computadores.

Desde 2014 contribuo com a vida das pessoas incentivando a entrarem na área de programação através do meu blog pessoal woliveiras.com.br, palestras, workshops e participação em

comunidades. Acredito cegamente que somente a educação pode mudar a sociedade em que vivemos, e por isso compartilho pensamentos e conhecimento pela internet afora.

Introdução

Começar na área de tecnologia não foi nada fácil para mim. Eu não tinha um amigo ou amiga, parente ou familiar que trabalhasse com tecnologia para me guiar. A falta de bons conselhos logo ali no começo foi algo que me atrasou um pouco a conquistar os meus objetivos como profissional.

Para você ter uma noção maior da minha história, eu venho de uma família periférica, onde a vivência profissional era baseada no que tinha para sobreviver. Eu curti bastante a minha infância, assim como qualquer criança que não tem noção das desigualdades sociais, mas a partir do momento em que eu precisava escolher uma profissão pensando em futuro a coisa complicou.

Conhecer o universo da programação mudou minha vida, pois tive acesso a pessoas que me apoiaram desde o começo, mesmo sem me conhecer. Esse apoio foi fundamental e tento repassar o que aprendi através da minha escrita no meu blog pessoal, o https://woliveiras.com.br, palestras ou através dos livros que escrevi.

Esta obra é uma coletânea de conselhos recebidos em determinados momentos da minha carreira ou prestados para alguém quando buscaram minha ajuda e eu sabia exatamente o que elas precisavam ouvir graças à experiência de vida em tecnologia. Muitos desses conselhos foram decisivos para conquistar uma carreira de sucesso, evoluir profissionalmente ou até mesmo pessoalmente, garantindo a integridade mental e física no mundo moderno e ambiente de constante evolução tecnológica, onde o trabalho se tornou algo tão essencial que já parece ser a única coisa que fazemos.

Desde 2014 venho aconselhando pessoas através do processo de mentoria nas comunidades de programação. Esse tipo de trabalho me trouxe muitas ideias e observações legais de se compartilhar com você através deste compilado mais organizado do que minhas ideias jogadas ao vento no blog e redes sociais.

A maneira como o conteúdo aqui é estruturado foi pensada para que você não precise ler todos os capítulos em uma sequência lógica. Cada parte pode ser lida em algum momento da sua carreira onde você precisa mais de uma informação do que de outra ou quando você sentir que precisa de um conselho amigo.

Talvez você já até tenha procurado um mentor ou uma mentora e não conseguiu isso a tempo. Aqui vamos encontrar experiências que detectei serem comuns às pessoas a quem dei mentoria e podem ser úteis para a sua caminhada profissional. Sinta-se recebendo conselhos do seu mentor e não hesite em me procurar nas redes sociais para conversar sobre qualquer assunto quando sentir que precisa da ajuda de alguém com mais experiência.

A quem se destina este livro

Esta obra foi escrita para pessoas de todos os níveis profissionais.

Visão geral

Procurei organizar os capítulos do livro do modo em que evoluímos em nossa carreira, passando pelas primeiras interações com a comunidade de programação, sobre desânimo na carreira, empregos ruins, até os pontos de vaidade que adquirimos com o passar do tempo.

Capítulo 1: partimos das nossas primeiras participações na comunidade de programação e como podemos cometer graves erros

em grupo caso não tomemos um certo cuidado nas interações de internet.

Capítulo 2: durante a nossa carreira, precisamos estudar e praticar muito, mas em alguns momentos deixamos de investir o nosso tempo em experimentação. Nesse capítulo refletimos sobre os porquês de isso acontecer.

Capítulo 3: desde a nossa infância, quase não recebemos estímulos para continuar aprendendo por conta própria, deixamos de experimentar e passamos a somente consumir verdades absolutas nos apresentadas em sala de aula. Visando melhorar um pouco a nossa trilha, nesse capítulo refletimos sobre o protagonismo no aprendizado.

Capítulo 4: como autodidata, cometi diversos erros de aprendizado e também deixei o ego subir muito a minha cabeça, me sentindo melhor do que outras pessoas por conta da facilidade em aprender por conta própria, mas o ego não é a única armadilha em que podemos cair como autodidatas e neste capítulo refletimos sobre isso.

Capítulo 5: a área de programação tem muito conteúdo disponível na internet, isso é bom e ruim ao mesmo tempo; precisamos aprender qual é o momento em que podemos seguir em frente.

Capítulo 6: as vagas de emprego mais nos assustam do que estimulam a candidatura. Neste capítulo refletimos sobre os requisitos desses anúncios e o que podemos fazer para conquistar a confiança para se jogar no mercado de trabalho.

Capítulo 7: algo que sempre me incomodou no discurso das empresas é a velha frase "trabalhe com o que você ama e nunca mais terá que trabalhar". Eu faço o que amo desde 2014 e em nenhum dia da minha vida profissional eu estive em um parque de diversões, por isso trouxe uma visão um pouco mais crítica sobre o que é trabalhar e a nossa relação com as empresas.

Capítulo 8: assim como o capítulo anterior, aqui eu também trouxe a criticidade de quem precisa observar o mercado de trabalho como uma pessoa profissional e não através da emoção, pois um emprego ruim pode iludir e prender você por um bom tempo por lá, até mesmo chegando ao burnout.

Capítulo 9: comunidades de programação são ambientes muito legais, mas precisam que cuidemos para o ciclo continuar. Nesse capítulo aprendemos como ajudar, mesmo se formos iniciantes.

Capítulo 10: vagas e formações exclusivas para pessoas de grupos sub-representados têm crescido no nosso mercado, nesse capítulo vamos refletir o porquê de isso acontecer.

Capítulo 11: a vaidade é algo muito humano e não vamos escapar de comportamentos ruins em nossa carreira como pessoas desenvolvedoras de software. Nesse capítulo vamos refletir sobre quais são os momentos em que o ego toma conta da nossa vida.

Capítulo 12: seja através das comunidades, vivência no trabalho ou educação formal, algumas coisas que reproduzimos são somente preconceito tecnológico, às vezes nunca nem utilizamos uma tecnologia ou ferramenta, mas como ouvimos várias pessoas falando mal, compartilhamos a mesma visão, então vamos refletir um pouco sobre como isso impacta a nossa carreira profissional.

Capítulo 13: O último capítulo é destinado a um pouco mais de motivação para a sua carreira profissional e, claro, meus contatos para o caso de você precisar.

Capítulo 1
Não seja babaca de comunidade

> Comunidades são demais, não deixe que o ego possa transformar este magnífico ambiente em lugares tóxicos

Começar o livro dando uma bronca pode parecer ruim, mas, para um bom convívio dentro dos grupos de programação ou tecnologias e para que eles continuem existindo, este alerta é extremamente importante.

Antes de trabalhar com programação, eu não tinha muita expectativa de qual seria o meu futuro. Não tinha aquele sonho de criança de trabalhar na mesma profissão dos pais, ser astronauta, jogador de futebol como alguns amigos ou outra conquista de vida profissional incrível (e muito menos o plano para chegar a qualquer lugar). Não imaginava qual caminho seguiria, tanto que, desde que comecei no mercado de trabalho, vivia trocando de emprego e até de área profissional assim como trocamos de roupa durante os dias. Já trabalhei como auxiliar de serviços gerais, conferente de logística, atendente de call center, auxiliar de funileiro, grafiteiro e outras coisas de que nem me lembro mais.

Gostei tanto dos grupos que rodeiam o assunto desenvolvimento de software, a dinâmica desses grupos, que foquei 100% do meu tempo em entrar nessa área por causa da motivação que essas pessoas me davam. Hoje tenho estabilidade financeira, boas condições de ajudar minha família e até de doar cursos e livros para pessoas que desejam entrar na área de programação e não possuem essa oportunidade. Inclusive, o meu primeiro livro, *O Universo da Programação*, é doado mensalmente através de uma iniciativa social pessoal no meu blog graças a tudo o que a comunidade de tecnologia me ajudou a construir.

Posso dizer com muito carinho que: "se não fossem as comunidades de programação eu nunca teria alcançado as coisas boas que alcancei em tão pouco tempo".

Porém, eu já fui um grande babaca e gastava parte dessa babaquice dentro dos coletivos.

Este capítulo é dedicado para que você entenda o que é uma comunidade de programação, qual a dinâmica destes grupos e alguns comportamentos tóxicos que podemos acabar levando para lá. É ideal que saibamos sobre isso, pois cuidar das comunidades garante que elas continuem existindo e, mesmo trocando as pessoas que estão ativas, o grupo continua funcionando e se transformando em locais cada vez mais legais. Além de entender o que é uma comunidade, vamos passear por alguns comportamentos ruins que cataloguei durante minha vivência nesses coletivos.

1.1 O que é uma comunidade de programação

Se você é uma pessoa totalmente nova nessa área, talvez nunca tenha ouvido falar sobre comunidades ou grupos focados em ajudar as pessoas de uma mesma profissão.

Comunidades de programação são grupos de pessoas que se juntam por um objetivo em comum: aprender ou praticar desenvolvimento de software, ou mesmo aprender ou praticar alguma tecnologia específica, como uma linguagem de programação (PHP, Java, JavaScript etc.), uma boa prática ou as novas tecnologias lançadas tão frequentemente (como os frameworks ou bibliotecas). É um espaço aberto para compartilhamento de conhecimento e incentivo da evolução da nossa profissão.

Aprendi bastante e muito rápido graças às dicas de pessoas que estão fazendo o bem para outras dentro desses grupos pela internet afora ou em encontros presenciais (conhecidos como *meetups*).

Essas pessoas foram importantes para remover várias barreiras da minha frente, foram como mentores e mentoras indiretas desde o começo da minha trilha como programador. Poderia até tentar citar os nomes aqui, porém foi tanta gente que me ajudou, que o livro seria somente dedicatória e teria mais de mil páginas.

Imagina se todas as profissões tivessem grupos de compartilhamento de conhecimento. A evolução social seria imensa e o tanto de novas formas eficientes de trabalhar é impensável. Nós temos um grande privilégio ao poder contar com esse tipo de comportamento na nossa profissão.

Nem tudo são flores

Mesmo que as comunidades sejam ambientes extremamente legais, sempre que existe um agrupamento muito grande de gente em um mesmo local aparecem as pessoas que não estão ali para agregar, mas para gastar seu tempo destilando ódio ou buscar alimento para o seu sadismo. Gente que só quer inflar o ego. Claro que também tem aquelas que até querem ajudar mas não sabem se expressar de forma menos rude ou só não têm empatia mesmo - muito provavelmente devido ao seu contexto de vida, isso não é um julgamento, somente a observação de um fato social.

Meu objetivo com este pensamento não é ofender alguém que tenha as más atitudes que eu vou comentar na sequência deste capítulo, mas sim para que façamos uma reflexão pessoal um pouco mais profunda antes de sair escrevendo qualquer coisa em comentários pela internet afora. A internet é um grande aglomerado de pessoas e pessoas são diferentes, assim como possuem traumas e necessidades que podem atacar gravemente através de um comentário seu.

Conforme artigo publicado no *Psychology Today*, um reconhecido portal sobre psicologia, o Ph.D. Glenn Geher constata um aumento de 59% de jovens em situação universitária com diagnóstico de algum transtorno psicológico na era da internet. Cito essa

informação para você avaliar o tamanho do impacto do que fazemos na rede mundial de computadores.

Eu continuo errando bastante e acredito que ninguém para de errar na vida, mas podemos nos policiar um pouco mais para que alguns erros não se tornem hábitos negativos em nossas vidas; em nossas relações sociais. Ainda me lembro de várias atitudes que eu tinha, continuo acompanhando os movimentos e vendo se repetir alguns comportamentos ruins por uma grande parte das pessoas desenvolvedoras.

As comunidades são cíclicas. O grupo de hoje não será o mesmo daqui há 5 ou 10 anos e talvez a nova formação pode repetir todos os erros que já cometemos. Talvez com este alerta, você possa ser a pessoa que vai fazer a diferença e influenciar positivamente nos grupos pelo mundo a fora.

1.2 Deixe-me usar o Google por você

É um dia normal. Estamos na internet compartilhando artigos, vídeos, podcasts e discutindo sobre tecnologia. Alguém pergunta algo extremamente importante para concluir uma tarefa ou resolver um problema em seu ambiente de trabalho em um fórum de comum acesso, como em um Stackoverflow (veja https://stackoverflow.com) da vida, e você envia aquele clássico link: **deixe-me usar o Google por você (https://lmgtfy.com)**.

> Não basta sermos babacas, precisamos demonstrar publicamente em todas as oportunidades possíveis, não é?

Não é raro encontrarmos uma pergunta muito "simples" de ser respondida se a pessoa que a enviou tivesse utilizado o Google como fonte de pesquisa em vez de usar o fórum com este objetivo. Então

alguém manda o *lmgtfy* para responder com sarcasmo e expressar um pouco de sua indignação com o comportamento de quem perguntou.

OK! Você pode usar o *lmgtfy* com seus amigos e amigas pela brincadeira, mas nunca com pessoas que você não conhece e vou explicar o porquê. Não sabemos quais são as dificuldades, muito menos o nível de conhecimento da pessoa que criou o tópico na internet. Acreditar que todo mundo na comunidade possui alfabetização digital 100% concluída, é Google Hacker ou tem as mesmas características/conhecimento técnico que você é um erro gravíssimo.

Não se esqueça: **a dúvida de alguém na comunidade é sempre importante** a não ser que a própria pessoa diga que não é, colocando um off-topic ou alguma tag similar combinada entre todos. Não é só por você ter seus 10 anos ou mais de carreira ou participar da comunidade há doze décadas que tem o direito de invalidar a necessidade de alguém. Nós nunca teremos esse direito.

Vai me dizer que você nunca fez perguntas “bobas”? Desde o seu primeiro dia estudando programação ou qualquer outra coisa que tenha aprendido em sua vida você já sabia pesquisar de **maneira eficiente** sobre aquele assunto? E mesmo que a resposta seja "sim" e você for uma pessoa extremamente fora da curva, será que todos têm que ser iguais a você?

Comunidades não são sobre você.

Quando iniciamos em qualquer área de conhecimento, não sabemos nem **o que** pesquisar na internet. Quanto mais saber como fazer boas pesquisas para encontrar o link matador que vai salvar o nosso dia. Às vezes, a resposta está a uma mensagem de erro na IDE de ser resolvida, mas estamos em uma imersão tão profunda no problema que nem lemos o erro piscando na nossa cara. Fora o fato de que, quando iniciantes, achamos que o erro sempre é nosso. Nossa lógica, nosso conhecimento, nossa escrita, e por vezes nem

olhamos o console da IDE ou navegador, simplesmente saímos procurando onde foi que fizemos m*rda.

Nenhuma pergunta é ruim. A pessoa tentou se comunicar. Se ela não soube se expressar bem, por qual motivo você não a ajuda a extrair mais informações do seu problema e formular melhor o seu tópico? Não é você que sabe mais ali? Talvez você não esteja realmente querendo ajudar, só aparecer como o engraçadão do grupo. Fazer uma gracinha e ter seus quinze minutos de fama ou, como costumamos chamar agora, ganhar biscoitos.

Reflita sobre isso: você, como membro de comunidade, não está ali só para usar e abusar do coletivo, está para fazer alguma coisa também. Você tem responsabilidade social dentro do grupo. Não é só ficar reclamando da moderação ou das questões que estão sendo publicadas e não ajudar em nada. Se a pessoa não explicou bem seu problema, ajude-a a se expressar melhor. Se você não tem paciência para ajudar, simplesmente não comente nada.

Quando alguém fizer uma pergunta muito simples, por exemplo, "o que é Nodejs?" em uma comunidade **de Nodejs**, onde a maioria já deveria saber pelo menos o que é isso... Pare, respire e pense antes de fazer uma babaquice. Se possível saia da frente do computador por um tempo. Você vive na frente da tela o dia inteiro, qualquer coisa vai disparar um comportamento negativo, ainda mais uma pergunta que pode incomodar.

Você pode fazer seu papel de pessoa útil dentro da comunidade e contribuir perguntando o seguinte:

- Você já pesquisou algo sobre o assunto? Se sim, quais foram os resultados conquistados?
- Por qual motivo você busca saber o que é *x*? Deseja criar algo ou aprender para trabalhar com isso?
- É seu primeiro contato com desenvolvimento de software?

Não saia julgando ninguém logo de cara. Por mais que exista um impulso dentro de nós chamado impaciência.

"Ah, mas as pessoas são preguiçosas", você pode refutar. E somos mesmo! Quem dera eu conseguisse aprender qualquer coisa sem precisar me esforçar tanto. Mas não cabe a nós presumir que uma pergunta muito simples ou mal formulada está descrita assim ou está no fórum somente por preguiça de formular melhor ou pesquisar na internet. Isso se chama projeção. Lançamos na outra pessoa o nosso juízo de valor pelo que nós faríamos, não pelo que está acontecendo na nossa frente.

Claro que existe um problema na minha premissa: a pessoa pode realmente nem ter pesquisado no Google sobre aquele problema antes de já sair perguntando na comunidade. Porém, como podemos saber se isso realmente aconteceu? Somente através de um comentário na internet não tem como.

Assuma que ela está buscando ajuda e é só isso que você precisa fazer: ajudar. Se não quiser gastar seu tempo e energia para isso, saia dali, não comente nada. Não precisa comentar. Ninguém obriga você a responder todos os tópicos dos fóruns.

Ainda sobre respostas em tópicos de fóruns

Quando eu comecei a me envolver mais profundamente na área, entrei em diversos grupos de programação aleatórios. Sério. Todo grupo que você imaginar, eu estava lá… PHP, Ruby, Python, Java, C#. Todos! No Orkut, Facebook, no LinkedIn, no Telegram ou nos sites específicos.

Certo dia eu fui a vários desses grupos e puxei a seguinte questão:

"Galera, como faço para me tornar um desenvolvedor x?"

Onde *x* pode ser PHP, Ruby, Java, Python, C#, C e JavaScript.

Eu fiz isso para levantar o máximo de informação sobre as tecnologias nas quais eu tinha interesse, como quantidade de material disponível, vagas na minha região etc. A resposta para essa dúvida me ajudaria a estruturar uma grade de estudos que eu iria

utilizar nos meses seguintes até conseguir arrumar um emprego na área.

Algumas respostas foram entristecedoras. Eu não tenho o link das respostas, pois faz muito tempo e o Orkut, uma rede social de grande sucesso na época de nossos ancestrais, já se foi. Porém, existiram pessoas que realmente queriam ajudar e me responderam com muito carinho e seriedade. Essas pessoas fizeram perguntas parecidas com as que eu comentei anteriormente e assim entenderam melhor o motivo pelo qual eu estava buscando apoio do grupo, tentaram entender meu contexto de vida e o que eu queria para o meu futuro profissional. Graças a essa galera, estou aqui hoje.

1.3 O que eu faço é sempre melhor do que o que você faz

Algo muito comum de se ver é a velha disputa do Dev Raiz vs. Dev Nutella, onde alguém diz: se x, você não é dev de verdade/raiz/bom/boa. Não importa o que seja.

Algumas pessoas têm tanta necessidade de se impor como melhor em alguma coisa, que perdem a noção do quão ridícula essa frase se torna quando começamos a buscar argumentos para reforçar o que foi dito. Dificilmente você encontra argumentos irrefutáveis sobre algo como:

- Usa Windows? Você não é programador(a)
- Se você não tem conta no GitHub você não é programador(a)
- Não é fluente em inglês? Você não é programador(a)
- Se você não tem faculdade, você não é programador(a)
- Se você não sabe o que é ASCII, você não é programador(a)

- Ainda não estudou sobre a arquitetura de Von Neumann? Você não é programador(a)

O maior problema nessas frases não é a falácia, mas no fato de que queremos que outra pessoa seja um espelho nosso. E, se ela não for, então ela não presta ou não é tão boa quanto nós somos (ou achamos que somos).

Pior do que isso é que, quanto mais "requisitos" a pessoa cumprir, mais o babaca vai trazer outros pontos para que ela não consiga completar a lista. É como ficar mudando a regra de um jogo com ele em andamento e você nunca conseguir ganhar. Gestores babacas também fazem isso com mulheres da área de tecnologia o tempo todo.

Será mesmo que só pelo fato de você usar Linux, não Windows, você tem mais capacidade de raciocínio lógico, conhecimento matemático, de eletricidade, eletrônica, física, conhece mais *design patterns*, domina mais de Arquitetura de Software, Ciência da Computação, Cyber Segurança e Engenharia do que outras pessoas, ou você só é mais uma bolacha no pacote e se acha a última?

Se você só quer que a outra pessoa se esforce para ser melhor no que faz, entenda que cada pessoa tem seu tempo, suas limitações e todas precisam de acolhimento para só então mudar suas vidas. Dividi-las em grupos de bons ou ruins por causa de uma coisa boba que só existe porque você quer não ajuda ninguém. Só deixa os espaços públicos menos seguros para iniciantes ou minorias sociais.

Espalhar o que comentei nos exemplos acima não faz com que as pessoas continuem se esforçando para serem melhores, só as coloca para baixo se sentindo mal e ainda pode ajudar a aparecer a famosa síndrome do impostor (onde a pessoa é, sim, boa no que faz, mas não consegue se enxergar como suficiente). Você pode ser responsável pela próxima pessoa a desenvolver ansiedade generalizada, depressão ou pelo próximo suicídio por pressão social.

1.4 Respeite as minas

É engraçado observar como alguns homens acham que o simples fato de a pessoa ser mulher a torna limitada dentro da área de desenvolvimento de software. Também é engraçado ver que o fato de ter mulheres ali faz com que a comunidade se torne um espaço de pegação na mente de alguns desses mesmos caras. É como se o grupo fosse um aplicativo de relacionamentos que está conectando pessoas que gostam da mesma coisa para se pegarem, não para compartilhar conhecimento.

Pode acontecer um *date* entre pessoas de comunidade? Lógico! E vai acontecer. Mas vai com calma, camarada. A pessoa está ali como profissional pedindo ajuda e deve ser respondida profissionalmente. Já vi casos de a mulher enviar uma pergunta e alguém comentar: "Nossa, como você é linda, nem parece programadora". O típico comentário que vai afastar a pessoa da comunidade, assim como atingir diretamente a autoestima das outras que estiverem no grupo. É como jogar uma bomba de mau cheiro em uma sala, as pessoas vão embora dali e só vai sobrar quem gosta desse tipo de ambiente ruim.

Em outros momentos aparecem perguntas muito bem formuladas, com exemplos completíssimos do que a pessoa fez, o que ela pesquisou e deu errado e mesmo assim algum homem aparece lá para refazer TODOS os passos deixando a dúvida: será que ele acha que a mulher está mentindo? Por que ele não tem o mesmo comportamento quando são homens perguntando?

O respeito em comunidades deve existir para todas as pessoas. Que tal cair na real de que não é por causa da sua genitália ou identificação de gênero que alguém ganha mais poderes do que as outras? Precisamos refletir e desconstruir o comportamento machista que reproduzimos o tempo todo com mulheres em tecnologia.

1.5 A internet conecta um mundo inteiro, saia do seu pequeno mundinho

Aparentemente é muito legal chegar a um espaço qualquer e zombar de algo que você considere engraçado, mas estamos na internet e isso torna o impacto do que falamos muito maior. Antigamente conversávamos em grupos de 3 até 8 pessoas. Agora, uma postagem em um fórum pode atingir milhares em questão de segundos.

A diversidade de vidas impactadas com o seu comentário é imensa. Você já parou pra pensar que pode estar zombando de alguém com distúrbios psicológicos que dificultam o entendimento de um tema e é por isso que ela manda essa pergunta "tão simples" na comunidade? Ou mesmo que, por alguma dificuldade de comunicação escrita, ela não desenvolveu a pergunta com o português mais correto?

Vamos pensar nestes poucos cenários que você quase não consegue perceber no dia a dia ou em postagens na internet:

- Dislexia
- Autismo
- TDAH
- Depressão
- Transtorno de Ansiedade Generalizada
- Transtorno Afetivo Bipolar
- Transtorno Opositor

No final da leitura deste capítulo, procure os nomes acima na internet e conheça um pouco do que pode ver por aí. Sua visão de mundo vai mudar bastante.

Pessoas com distúrbios ou com alguma deficiência podem trabalhar do seu lado. Alguém com transtorno de ansiedade generalizada e que já teve ideação suicida por causa do impacto da internet em sua vida escreveu o livro que você está lendo neste momento.

Conclusão

Este capítulo é um apelo ao pensamento autocrítico. Espero ter colocado cenários diferentes em sua vida para que os problemas expostos aqui não sejam reproduzidos por você. Antes de receber dicas e levar muita bronca de pessoas próximas eu reproduzia muitos comportamentos ruins. Nós somos humanos e errar é quase inevitável. O importante é sairmos daqui com uma atitude mais positiva e não mais cometer os mesmos erros.

Analise-se um pouco e pense muito bem antes de dizer algo na internet para que não sejam suas palavras as culpadas por um pedido de demissão, desistência da profissão ou, levando ao extremo, um suicídio.

Use a autocrítica ao seu favor e #SoNaoSejaBabaca.

Referências

THE Mental Health Crisis is upon the Internet Generation. *Psychology Today*. Disponível em https://www.psychologytoday.com/intl/blog/darwins-subterranean-world/201707/the-mental-health-crisis-is-upon-the-internet-generation

CAPÍTULO 2
Por que você desistiu de ser cientista?

Eu não sou psicólogo, estou bem longe disso. Mas há algum tempo observo nosso comportamento, mais especificamente as pessoas da área de desenvolvimento de software, que, afinal, é o espaço social onde estou inserido. Desde que comecei a pensar sobre comportamento, percebi um padrão que se repete na maioria de nós:

> Ao crescer, abandonamos a "profissão" que nasce conosco: a profissão **CIENTISTA**.

Deve existir uma razão psicológica, sociológica, filosófica para isso, não sei. Eu não posso dizer com tanta certeza, pois nunca achei nenhum estudo sobre o tema (talvez eu nem saiba como procurar). Porém, neste momento eu quero somente deixar minha reflexão para você, que desistiu de ser cientista e não sabe mais experimentar as coisas. E, talvez, conseguir convencer você de que resgatar a nossa criança interior nos torna melhores pessoas desenvolvedoras de software.

Pode soar como papo de *coach* quântico ou jovem místico, mas eu realmente refleti por um bom tempo sobre as questões que vou trazer aqui.

Neste capítulo você encontra um pouco sobre o que eu penso a respeito da ciência, sobre como aprendemos na infância e alguns pontos sobre ser criança que nunca deveríamos ter abandonado.

2.1 O que faz uma pessoa cientista?

Muitos de nós enxergam uma pessoa cientista como o ser mais inteligente do universo, pesquisando a solução para os maiores problemas da humanidade ou enviando pessoas para outros planetas. Acho que esse primeiro trecho da definição da Wikipedia sobre cientistas é a melhor que eu poderia utilizar para explicar meu ponto de vista:

"Um cientista, em um sentido mais amplo, refere-se a qualquer pessoa que exerça uma atividade sistemática para obter conhecimento." (Wikipedia. https://pt.wikipedia.org/wiki/Cientista)

Pense que essa atividade sistemática não significa seguir um modelo existente, mas criar o seu próprio sistema e repeti-lo até conquistar o conhecimento desejado. Já reparou que caímos várias vezes até conseguirmos nos levantar ou fazemos vários sons antes de conseguirmos dizer uma palavra bem simples?

Todos nascemos assim, com essa característica exploradora de busca pelo conhecimento de qualquer forma. Crescemos buscando experiências, querendo aprender, explorando as coisas ao nosso redor. Nós nos machucamos, machucamos os outros sem querer, ganhamos algumas mordidas de animais... Isso é experimentar.

Mas, olhando para dentro agora, quanto tempo faz que você não tenta algo novo? Quando foi a última vez que você criou algo do zero sem olhar uma solução pronta de outra pessoa no GitHub ou em buscas no Google (não estou falando de escrever código igual, mas de ter uma ideia criativa para resolver um problema computacional)?

Na infância fazemos muito disso: experimentamos, criamos, buscamos. Porém, o tempo mata um pouco da nossa capacidade, talvez essa energia, mata a criatividade e pessoas desenvolvedoras de software precisam ser criativas.

Quando deixamos de ser cientistas?

Sir Ken Robinson, um dos grandes nomes na área de estudo sobre a criatividade, diz que “as escolas acabam com a criatividade das crianças”. Eu concordo e muito com ele. Para mim, o sistema educacional é falho em estimular a criatividade das crianças, pois está tão preocupado com padrões da sociedade, MEC (para escolas brasileiras), ego do próprio professor, dentre outras coisas, que não estimula a criatividade das nossas crianças.

Somente ensinam o que precisam para passar em um vestibular, aprender a ler, escrever e falar. Não estimulam as diferentes inteligências que temos, nem que continuemos criando coisas e sedentos por conhecimento. É como estar em uma fábrica onde colocam você dentro de um molde e soltam para o mundo para ser só mais um por ali.

Quando eu, meus amigos ou pessoas que eu questionei sobre este assunto para a escrita deste capítulo estávamos nas escolas, só queríamos que acabasse logo. Não culpo a escola (instituição) ou todos os professores, nosso problema é social. A sociedade foi se limitando com o tempo até chegarmos ao lugar onde estamos hoje. As escolas só são um reflexo disso tudo o que criamos ou que deixamos acontecer, elas estão preparadas para o que aquela sociedade precisa. Será que a nossa sociedade não precisa mais de criatividade?

Se você já passou pela escola, vota e contribui com a sociedade de alguma maneira (impostos, por exemplo), você também tem parte de "culpa" nessa história, você é parte da sociedade, você é parte do problema. Não vamos jogar a responsabilidade somente nas instituições de ensino, pois existem professoras que lutam dia após dia para tentar estimular as crianças de maneira diferente da que lhes é ordenada e em sua maioria essas professoras são boicotadas pelo sistema ou pela falta de recursos.

Os pais, para um crescimento utilitarista, por vezes não querem que as crianças aprendam algo que **elas** buscam por conta própria, mas que estudem e tenham foco somente no que os pais supõem que

será útil no futuro baseado em suas intuições e história de vida. Às vezes acertamos, mas nem sempre nossas intuições estão certas. Muitos pais de filhos e filhas da década de 90 estimularam que a busca fosse pela área de fábricas de automóveis, por exemplo, porém o mercado de tecnologia dominou o mundo e hoje temos vagas sobrando para trabalhar nesta área e poucas fábricas de automóveis com oportunidades.

Para sermos bons desenvolvedores e desenvolvedoras, não adianta somente bater tecla o dia inteiro, escrever código na linguagem de programação mais rápida, mais bonita ou mesmo ficar lendo requisitos e cuspindo código na tela como um robô. Você precisa da criatividade para criar boas soluções ou soluções que ninguém pensava em desenvolver. Isso é o que nos difere no dia a dia. Você terá uma caixa de ferramentas totalmente recheada de padrões de projetos, algoritmos e estruturas de dados, convenções e boas práticas, e o limite de quão impressionante pode ser o que você fará com isso está em sua capacidade criativa de sair fora da caixa.

2.2 Deveríamos nos espelhar nas crianças

Esquece aquele cientista famoso que você tanto admira, a pessoa desenvolvedora mais fera do seu trabalho ou aquela que é referência na sua área por ter feito 30 palestras em um ano, ter criado um blog/site famoso ou que vive ditando regras toscas sobre programação no Twitter! Esquece.

Espelhe-se nos seres mais cheios de curiosidade que tem para conhecermos: os gatos! Brincadeira, as crianças. Até agora eu comentei como perdemos nossa capacidade criativa, mas não sabemos se dá para voltar a ter aquilo. Talvez consigamos um pouquinho do que já tivemos se analisarmos alguns pontos, que comento logo a frente, como curiosidade, insistência e a coragem.

Curiosidade

Um traço da personalidade de cientistas (profissionais mesmo) é que são pessoas extremamente curiosas por natureza. Parece algo mágico ou uma característica única desse perfil profissional, mas não é bem assim. A profissão acaba forçando e criando o hábito da sempre busca por mais conhecimento, informações ou novas soluções para problemas antigos.

Você também ainda é uma pessoa curiosa, mas acabou deixando morrer a força de vontade de descobrir o mundo a sua volta e tem se esforçado em saber quem vai morrer na novela/no filme/na série, por que ela morreu, o que aconteceu, você quer saber o que seus amigos estão fazendo nas redes sociais, quer saber quem está namorando com quem, só pensa no emprego, e por aí vai…

Não quero dizer que isso seja inútil, afinal precisamos de distrações e momentos de descontração em nossas vidas e de trabalhar para alimentar os nossos animaizinhos. Porém, seria bom se reservássemos um tempo para exercitar mais a curiosidade a fim de adquirir algum conhecimento legal ou experimentar algo novo. Lembra quanta coisa aleatória você aprendia na infância? Conhecimentos gerais, habilidades manuais, tem um mundo inteiro para descobrirmos fora da nossa bolha.

Não quero que você pense que eu estou tentando limitar você a buscar conhecimento no que EU acho útil, pelo contrário, quero dizer que você deveria buscar coisas legais que podem agregar mais em sua vida, como ler/consumir conteúdo sobre psicologia, administração, comunicação, se exercitar, aprender a construir objetos com madeira e assim vai.

Quando foi a última vez que você parou para pesquisar sobre como alguma coisa que você utiliza funciona? Você sabe como funcionam os chips que são usados no seu celular? Sabe qual é o modelo de processador do seu celular? Exemplos "simples", mas que nem todo

mundo teve empenho para descobrir como funcionam. Comece pequeno, por itens a sua volta.

Insistência

As crianças não desistem tão fácil de aprender algo novo.

Tente enrolar uma criança, sem responder a uma pergunta que ela fez, e verá como elas conseguem nos tirar do sério questionando mais e mais sobre aquele assunto até não termos mais energia para continuar com a enganação. Mas não é só no quesito conversa que elas nos superam, elas caem e se levantam para continuar tentando, nós, depois de adultos, acabamos perdendo muito dessa característica.

Só de pensar na dificuldade de procurar material para estudar algo a gente já se cansa. Parece que vamos perder muito tempo se investirmos energia em algo diferente do que está utilizando no dia a dia — talvez sim, mas talvez seja muito recompensador para sua criatividade. Isso acontece por causa do utilitarismo, só queremos fazer o que é útil no curto prazo e não enxergamos novas possibilidades no que fazemos.

Invenções

Se você entregar um problema para uma criança resolver e ela não souber a resposta, ela tem a opção de dizer que não sabe ou inventar algo extremamente louco para resolver isso. Normalmente elas optam pela segunda opção. É até engraçado.

Pergunte algo extremamente adulto para uma criança curiosa, como "você vai declarar o seu imposto de renda esse ano?" ou "você acha que o aquecimento global é algo ruim para nós, seres humanos?". Eu fiz este teste com crianças de 7 anos e foi muito engraçado.

Tudo bem que, com o tempo e a idade, você não pode simplesmente falar algo que não sabe, não pode ficar inventando histórias para os outros, porém você poderia colocar a mão na

massa e inventar algum software com sua criatividade, algo só por diversão, por experiência pessoal ou para aprender algo novo. E disso ainda pode surgir um produto que auxilie outras pessoas.

Nós, como pessoas desenvolvedoras de software, carregamos um poder imenso de mudar vidas, mas estamos tão presos na rotina que não enxergamos isso nem mesmo para construir ferramentas que nos auxiliariam.

O medo

Depois de um tempo, com os pais, escolas e sociedade impondo as regras arbitrárias para o convívio em grupo, as crianças acabam parando de realizar os hábitos legais que tenho comentado até aqui. Mas até certa idade elas ainda não têm medo de errar.

Você já deve ter perguntado em algum fórum de tecnologia:

- Gente, como eu faço esse efeito na minha interface?
- Galera, como eu crio isso?
- Pessoal, me ajuda a fazer meu trabalho?

Mas essas perguntas deveriam ser respondidas por você mesmo! Você deveria testar, experimentar as coisas, tentando inventar. Ou, pelo menos, ter pesquisado mais no Google. Depois era só buscar uma maneira de melhorar o que você criou e lançar. As pessoas que lhe respondem são as que experimentaram, buscaram e adquiriram assim mais conhecimento.

Talvez você pensou: “Ah, mas precisamos ser ágeis no dia a dia, não dá para ficar reinventando a roda”. Concordo 100% contigo, no dia a dia de quem desenvolve, não há tempo de ficar experimentando muitas coisas, mas desde quando isso nos limita para aprendermos algo novo fora do trabalho? E eu não estou falando para reinventar a roda, mas aprender por conta própria como as construções funcionam por baixo dos panos.

Na faculdade, curso ou individualmente, você aprendeu C e Java ou outra linguagem, isso o limita a não buscar aprender novas linguagens depois? As pessoas autodidatas vivem experimentando e em determinado ponto param por algum motivo. Isso deveria ser revisto! Por que se limitar?

E experimentar algo novo, praticar, brincar com uma nova tecnologia ou inventar uma coisa nova com suas ideias não é o mesmo que ser *workaholic*. Estou tentando dizer que você pode brincar com o que você faz, sim, mas fique de olho quando até mesmo se divertindo você está focando em vida profissional em vez de só experimentar.

Nossa profissão não nos limita a uma empresa, nós podemos nos divertir fazendo o que fazemos no serviço e também podemos nos divertir fazendo algo legal/aprendendo algo legal em casa.

2.3 Sair da zona de conforto

Talvez o ponto crucial deste capítulo seja o que vou falar agora:

> **VOCÊ PRECISA ABANDONAR A PREGUIÇA!**
>
> **Isso está matando sua criatividade.**

Hoje você já não é mais uma criança, não tem mais a escola limitando ou seus pais impondo o que você precisa fazer/aprender, qual caminho seguir. Está na hora de você tomar as rédeas da busca por conhecimento. Vai logo aprender algo novo ou vai colocar um pouco de terra na boca e conhecer o mundo!

Expandir a nossa zona de conforto é algo maravilhoso. Ao invés de se acabar de estudar simplesmente para ter mais o que falar, aprenda até que aquilo se torne tão simples para você que vire só uma camada da sua zona de conforto.

Talvez esse não foi o melhor texto que você leu ou vai ler sobre criatividade e inteligência, mas espero ter mudado um pouco sua mente e estimulado você a arriscar mais.

Conclusão

O mundo não é amistoso com o nosso processo criativo. A industrialização, o Fordismo, trouxe um modelo de trabalho repetitivo e pouco criativo para entregar o máximo de itens iguais em pouco tempo. Isso também chegou nas nossas entidades de ensino, sejam públicas ou privadas, cujo objetivo é formar o máximo de pessoas o mais rápido possível, com notas específicas nas matérias "mais importantes", como matemática ou português e abandonam os esportes e artes. O nosso futuro muitas vezes não é realmente levado em consideração, mas sim uma grade genérica que atenda à sociedade atual.

Claro que ter uma grade específica para cada grupo de alunos e alunas com necessidades diferentes poderia dar muito trabalho, não sei. O fato é que em algum momento precisamos voltar às bases e buscar novamente a nossa criatividade infantil, nossa vontade de conhecimento sem medo de explorar o máximo possível.

Depois dessa leitura, que tal aprender algo novo? Abre a sua rede social preferida e procure: DIY (*Do it yourself*, "Faça você mesmo").

Capítulo 3
Assumindo o papel de protagonista em nossa aprendizagem

> Conhecermo-nos melhor nos ajuda a aprender mais. Tomar as rédeas dos nossos estudos pode mudar o rumo da nossa vida.

Não temos dúvidas de que o estudo, o conhecimento, muda as nossas vidas. Algum tempo atrás (mais especificamente em 2009) eu não tinha nenhuma perspectiva de futuro, do que eu faria da vida, profissão ou nada do tipo. Nunca fui estimulado a pensar nisso. Porém, ao me envolver com tecnologia, fui apresentado a uma nova maneira de pensar e agir. Estudar por vontade própria, não para passar em uma prova, transformou e muito o que eu sou.

Foi naquele momento que eu percebi que precisaria mudar um pouco a maneira como eu aprendia, afinal de contas, não aprendemos a aprender, aprendemos a seguir regras e passar em provas. Mas foi naquele mesmo momento que me deparei com o maior problema que poderia enfrentar como pessoa autodidata: o medo de estar fazendo algo errado.

Esse medo limita muito as nossas tomadas de decisão. O modelo social em que vivemos, baseado em julgamento e aparência em redes sociais, proporciona mais receio ainda de tentar algo e não conquistar o sucesso. Parece que estamos em um jogo onde, se você não realizar sempre o que planejou, então é fim de carreira para você.

As barreiras mentais também chegam à nossa capacidade de aprendizado, como comentei no capítulo anterior. E por isso a onda do "método de aprendizado" que recebemos nas escolas tradicionais, onde temos uma pessoa protagonizando o papel de

transmissora de conteúdo e nós como receptores, vai nos distanciando cada vez mais daquilo que podemos realizar se desbloquearmos nossas habilidades de autoaprendizado.

E assim a vida vai seguindo desde o primeiro ano do pré-escolar até a educação superior: tem medo de errar, encontra uma fórmula que funciona para conseguir um emprego, segue a fórmula, tem medo de errar, encontra outra fórmula, repete... Só que aprender não precisa e nem deveria ser sempre assim.

Com o tempo, eu aprendi a assumir o papel de protagonista em minha aprendizagem e quero compartilhar sobre isso com você neste capítulo. Vamos passar por tópicos sobre como funciona o aprendizado e em seguida refletir como podemos assumir as rédeas do nosso processo, aprender a utilizar a tecnologia para investimento em conhecimento e algumas dicas de como estruturar um plano de estudos (somente como exemplo).

3.1 Nossa infância

As primeiras pessoas a protagonizarem o triste papel de travar nossas mentes para sermos somente receptores de conhecimento são nossos professores, professoras e a diretoria do pré-escolar que seguem um modelo antigo de ensino, que precisa muito de uma atualização. Não só pela evolução tecnológica e a perda de tempo em não utilizar os recursos em sala de aula, mas também porque existem outros métodos baseados em exploração e autogestão que são muito melhores.

"O Método Montessori é o resultado de pesquisas científicas e empíricas desenvolvidos pela médica e pedagoga Maria Montessori. É caracterizado por uma ênfase na autonomia, liberdade com limites e respeito pelo desenvolvimento natural das habilidades físicas, sociais e psicológicas da criança.

De acordo com sua criadora, o ponto mais importante do método não é tanto seu material ou sua prática, mas a possibilidade criada pela utilização deste, de se libertar a verdadeira natureza do indivíduo, para que esta possa ser observada, compreendida, e para que a educação se desenvolva com base na evolução da criança. A criança é o centro do método montessoriano e o professor tem o papel de acompanhador do processo de aprendizado. Ele guia, aconselha, mas não dita e nem impõe o que vai ser aprendido pela criança." Fonte: Wikipédia (https://pt.wikipedia.org/wiki/M%C3%A9todo_Montessori)

Na Escola da Ponte, uma instituição localizada em Porto - Portugal, é referência na autonomia das crianças e na construção de mentes pensante e brilhantes graças ao seu método de ensino. Segundo o idealizador da escola, José Pacheco:

Como diria Paulo Freire, ‘aprendemos uns com os outros, mediatizados pelo mundo’. Freire diz-nos que a aprendizagem não está centrada no professor, nem no aluno, que aprendemos na intersubjetividade. A educação é um direito também subjetivo. Todo ser humano tem direito a desenvolver seus talentos. No desenvolvimento de projetos, acontecerá não o consumo acéfalo de currículo, mas a produção de conhecimento e de currículo. A partir de sonhos, necessidades e desejos de cada ser humano e integrando conteúdos, competências e capacidades de uma base curricular, visa-se estimular talentos e cultivar os dons de cada sujeito aprendente.

Hannah Arendt dizia que, quando alguém nasce, algo novo acontece no mundo. Dado que um ser humano é único e irrepetível, no desenvolvimento do currículo da subjetividade é respeitada a especificidade do seu repertório linguístico e cultural, dos seus estilos de inteligência predominantes e do seu ritmo de aprendizagem. Fonte: Aliança Pela Infância (http://aliancapelainfancia.org.br/inspiracoes/individualidade-respeito-e-autonomia-como-funciona-a-escola-da-ponte-em-portugal)

Em escolas particulares bem caras, já se utiliza o método Montessori, por exemplo. Mas isso está longe de ser a realidade na grande maioria de escolas públicas, onde o modelo que está sendo aplicado atualmente é o da cartilha educacional, na qual nem mesmo o professor tem autonomia sobre o que ensinar. Professores que, devido à falta de liberdade educacional, já começam a nos colocar em um molde, onde: você vai entrar em uma sala, ouvir uma pessoa que está lá na frente e detém toda a experiência de vida e teoria necessária para o seu futuro, e em seguida fazer tudo o que ela mandar, sem questionar seu direcionamento. Se questionar, corre o risco de levar uma advertência ou ser humilhado(a) em sala de aula, na frente de todos os seus amigos e amigas.

Quem nunca achou que professores(as) não podem errar? Em determinados momentos de nossas vidas achamos até estranho quando vemos nossos docentes cometendo uma gafe. Isso acontece porque crescemos acreditando que eles e elas sabem tudo, dominam as artes ocultas da busca por conhecimento e é por isso que estão à frente da nossa educação formal.

Você consegue se lembrar um pouco (nem que seja bem pouquinho) da sua infância e identificar problemas nos modelos de ensino utilizados hoje em dia?

A criança cresce pensando que para aprender é preciso:

- alguém que vai guiar o aprendizado, entregar um currículo pré-montado e passar o conhecimento
- uma grade de estudos predefinida por outra pessoa
- avaliações puramente teóricas para "confirmar" se ela aprendeu

O problema neste método de aprendizagem é que o protagonismo está invertido. Crescemos **terceirizando** nossas tomadas de decisão.

A criança aprendeu de verdade ou só seguindo regras? Provavelmente aprendeu algo sim, mas ela não está sendo protagonista e muito menos aproveitando tudo o que ela poderia aprender se fosse, de fato, o foco dentro da sala de aula. Este conteúdo será perdido assim que ela sair da escola (e não estou falando que o conteúdo é desnecessário, o método é que já não se aplica mais).

As pessoas crescem passando pelas mesmas etapas na vida e, quando chegam a certa idade, umas aprenderam muito mais que as outras, mesmo alguns tendo as mesmas capacidades ou oportunidades (levando em consideração uma comparação justa entre essas pessoinhas). Isso é a maior prova de que não estamos aprendendo corretamente na infância.

3.2 Depois da primeira infância

Depois da adolescência e durante a vida adulta, esse molde já nos tornou um cidadão padrão. Agora não temos mais a capacidade de simplesmente pegar um livro e aprender algo, só o vamos fazer se alguém indicar. Só vamos explorar um conteúdo se alguma referência de mercado dizer que precisamos disso. Os fóruns de tecnologia estão cheios de perguntas se "vale a pena investir em x ou y" para validar suas tomadas de decisão. Olha o medo de errar aí.

Claro que esse medo envolve muita coisa, como a oportunidade financeira de investimento em educação, mas muito conteúdo da nossa área está gratuito na internet, não é necessário investir dinheiro em um primeiro momento.

Isso acontece porque estamos enviesados a imaginar que para aprender precisamos que alguém explique exatamente o que precisamos aprender e pronto. Não temos a prática de ir atrás do conhecimento por conta própria.

Tudo bem precisar de alguém para aprender, isso não é ruim. Às vezes não temos tempo para absorver um assunto mais denso e alguém já compilou esse conhecimento para nós e isso é ótimo no ritmo intenso que levamos nossas vidas hoje em dia, tendo que trabalhar, estudar, cuidar da casa, cuidar da saúde, de filhos, de animais de estimação, dos parentes etc.

Claro que existem ótimos cursos que estimulam seu aprendizado com as mais conhecidas técnicas de aprendizado já pesquisado pela área de educação, como os cursos da Alura, do mesmo grupo da editora deste livro, mas, a grande maioria trata de um modelo padrão aplicado para uma grande massa. Um curso bom deve estimular sua capacidade de buscar por si só, e não existir ali simplesmente porque teve mercado, gente procurando conhecimento sobre um assunto, e alguém foi explorar esse

mercado para lucrar com a nossa falta de capacidade de tomadas de decisão.

Ou seja, quando adulto, repetimos os mesmos processos e nada mais. Só que agora envolve nosso dinheiro e um tempo muito mais precioso do que na infância.

O que é protagonizar nosso aprendizado

Assumir as rédeas do aprendizado não é fácil. É uma quebra de paradigma muito grande por já estarmos acostumados(as) com o modelo atual. Porém, não é impossível. Daqui para a frente, vamos pensar o que é ter autonomia quando você pensa em estudar. Vamos analisar um pouco mais como o conteúdo está estruturado e por que ele está assim, e talvez possamos montar por conta própria nossa grade de estudos com base no que precisamos agora, não em possibilidades futuras que só nos dão ansiedade e medo do que pode acontecer.

Quando temos um professor ou professora lá nos ensinando, qual foi o grande trabalho dessa pessoa (além de fazer faculdades e cursos da área que ensina, claro)? A resposta é: estruturar o conhecimento em tópicos que devem ser aprendidos sobre o assunto ao qual ela ministra a aula, organizar esse passo a passo em um fluxo lógico com dificuldade gradual, aprender o máximo possível sobre esses temas, buscar referências que reforcem o que será ensinado e, enfim, ensinar.

Isso é o que nós deveríamos conseguir fazer sozinhos quando desejamos aprender algo novo, mas não temos o hábito, não aprendemos ou temos preguiça de arregaçar as mangas e fazer tudo isso — não existe problema em ter preguiça, você pode, sim, sentir isso, ok?

Para que tenhamos vontade de aprender de fato, além de buscar criar o hábito de aprender algo novo sempre, precisamos buscar uma motivação e nada nos motiva mais do que a nossa autonomia. O poder de estar à frente das decisões, construir as pontes e chegar ao nosso objetivo final. Isso realmente é muito bom para o nosso cérebro.

Faça um teste rápido antes de continuar essa leitura: pesquise no seu navegador "como a autonomia nos motiva?". Você verá resultados que vão desde métodos de ensino de idiomas até a produtividade no trabalho.

3.3 Tecnologia como ferramenta de aprendizagem

Agora que sabemos o que fazer para ter uma capacidade de aprendizado maior e mais motivação para seguir um plano, podemos usar essa ferramenta maravilhosa que é a internet para absorver muito mais. Temos quase tudo de que precisamos para adquirir conhecimento na rede mundial de computadores e a maioria, de graça.

Se você quer conhecer algo, basta pesquisar na internet e pronto. Hoje podemos conhecer culturas sem precisar sair de casa. Se você quer aprender a chegar a um lugar, basta usar um aplicativo de mapas da vida. Se você quer aprender outro idioma, temos ferramentas para isso. Enfim, tudo de que precisamos está aqui, vamos aprender a usar.

O que você vai precisar agora é estruturar o conteúdo. Criar um fluxo lógico com aquela dificuldade gradual que eu comentei (não adianta tentar criar um foguete começando pelo lançamento), estudar muito e compartilhar o que você aprendeu. Isso, além de estimulante, democratiza o acesso a informação e viabiliza que

outras pessoas assumam o protagonismo do seu aprendizado através do seu compartilhamento.

> Para não ficar só em minhas palavras, procure depois sobre o método de estudos mais fantástico que existe: o método Feynman.
>
> Talvez seja o jeito mais simples de aprender qualquer coisa. Richard Feynman foi um físico norte-americano do século XX que, em 1965, ganhou o Prêmio Nobel por ser precursor nos estudos de eletrodinâmica quântica. Feynman acreditava que poderia aprender sobre qualquer assunto, desde que fosse ensinado utilizando termos simples.
>
> São quatro passos simples de absorção do conteúdo, onde passamos pela escolha do assunto, ensinar sobre este tema para alguém ou simular uma aula, identificar as falhas do nosso entendimento sobre o assunto, revisar tudo e simplificar. A técnica consiste em deixar o conhecimento (de qualquer tema) tão simples que uma criança possa aprender.

Por enquanto vamos aprender como eu estudo atualmente, baseado em alguns conteúdos que estudei até chegar nesse modelo. E, para que o protagonismo seja seu, leia como eu faço e tente implementar da sua própria maneira em sua rotina.

3.4 Como estruturar o conteúdo

Antes mesmo de começar a estudar qualquer coisa você deve estruturar seu raciocínio e os tópicos que vai aprender. Como um índice de livro ou grade curricular que encontramos nos sites das universidades/cursos.

Pegue um assunto que deseja aprender, busque o máximo de informações sobre ele. Sério, o máximo de conteúdo bem por cima para entender o que é que estamos buscando. Você vai perceber um padrão de informações que se repetem nos artigos, livros e outras fontes de informação.

Tudo o que for repetitivo entre essas fontes é onde devemos focar de começo. Tudo o que não for repetitivo vamos colocar como tópico secundário sobre esse assunto.

Imagina o seguinte cenário: eu gostaria de aprender JavaScript.

Em todo lugar sempre encontramos:

- o que é a linguagem JavaScript
- como a linguagem funciona
- escrevendo seus primeiros programas (hello world)
- algoritmos com JavaScript
- front-end/back-end com JavaScript

Então, o que fazemos, é pegar as repetições, colocar em alguma ferramenta para estruturar nosso raciocínio (pode ser um mapa mental, uma agenda ou uma planilha, por exemplo) e em seguida nos aprofundar em cada item por si. Se possível podemos escrever rascunhos, artigos em nossos blogs ou qualquer atitude que nos ajude a guardar o conhecimento.

3.5 Construindo um mapa mental

Existem diversas maneiras de construirmos um mapa mental eficiente. A minha favorita é utilizando os aplicativos de celular ou softwares que nos ajudam nessa missão. Eu gosto dos aplicativos porque você não precisa se preocupar em apagar o desenho ou mesmo carregar um caderno em todos os lugares. Nós já levamos os celulares para qualquer canto.

Mas, antes mesmo de instalar um app no seu celular, vamos entender o que precisamos pensar para construir o nosso mapa. Vamos continuar com o nosso exemplo de aprendizado sobre JavaScript. Em um mapa, nós colocamos uma palavra central, um termo que representa algo muito importante que queremos lembrar ou, como no nosso caso, uma ideia a estruturar. Ao redor deste termo, vamos puxando arestas, como galhos de árvores, que ligam outras palavras relacionadas e isso nos ajuda a ter uma visão maior de tudo o que vai acontecer.

Com a missão "aprender JavaScript" em mãos, podemos pensar o seguinte:

- Aprender JavaScript
    - Como JavaScript funciona?
    - O que é JavaScript?
    - Para que serve JavaScript?

Já temos, então, o termo central e algumas arestas para o nosso mapa. Depois, vamos aprofundando ainda mais.

- Aprender JavaScript
    - Como JavaScript funciona?
        - Entender o que é um compilador
        - Entender como o navegador compila o JavaScript
    - O que é JavaScript?
        - Entender o que são linguagens de programação
        - Entender como um programa de computador funciona
    - Para que serve JavaScript?
        - Entender como funciona esse trem de front-end e back-end
        - Entender como usar JavaScript no meu dia a dia

E, de acordo com que vamos buscando mais referências, o nosso mapa vai crescendo.

Isso poderia ser um desenho em uma folha de caderno ou em um A4. Mas, como comentei, eu prefiro os aplicativos para ter mais

liberdade na construção dos mapas. Costumo utilizar o Mindmeister (mindmeister.com) tanto no celular quanto no navegador.

Tente realizar o exemplo acima no Mindmeister. Copie o termo central e vá puxando as arestas para cada item que eu coloquei na lista.

Criando um fluxo lógico

Com o mapa mental ou rascunhos em mãos já será possível começar a estudar. Porém, pode ser bem útil, devido ao nosso hábito já enraizado de aprendizado, criar um fluxo lógico de por onde começar e para onde ir. Não tem ninguém para nos dizer isso e o medo vai bater.

Este fluxo nada mais é do que uma grade do conteúdo, começando do básico e subindo o nível gradualmente. Essa parte de subir de nível é extremamente importante para que não comecemos pelo lado errado e acabemos desistindo de estudar. Imagine o fluxo lógico como um índice de livro ou a grade de um curso mesmo. Você vai escrever sua própria grade de estudos e, para facilitar a nossa vida, podemos, inclusive, olhar os livros e nos basear neles. É o caminho mais eficiente que eu encontrei para a construção de um mapa de estudos.

Algo também importante em nossa atividade deve ser a repetição espaçada. Repetir/relembrar um conteúdo que foi apresentado no começo dos estudos é extremamente importante para reforçar para o nosso cérebro que aquilo é importante e por isso deve ser armazenado com carinho em nosso subconsciente. Para isso, podemos colocar atividades similares ao que foi realizado no começo lá no meio da nossa grade de estudos. Lembre-se: repetir conteúdo não é interessante, você pode repetir uma atividade prática, mudando alguns parâmetros que façam você tomar decisões diferentes e aprender mais.

Vamos pegar o nosso mapa mental:

- Aprender JavaScript
  - Como JavaScript funciona?
    - Entender o que é um compilador
    - Entender como o navegador compila o JavaScript
  - O que é JavaScript?
    - Entender o que são linguagens de programação
    - Entender como um programa de computador funciona
  - Para que serve JavaScript?
    - Entender como funciona esse trem de front-end e back-end
    - Entender como usar JavaScript no meu dia a dia

Vamos aumentar a quantidade de itens dentro deste mapa.

- Aprender JavaScript
  - Construir um TODO List com JavaScript
  - Programação Funcional com JavaScript
  - Programação Orientada a Objetos e JavaScript
  - Introdução a programação com JavaScript
  - Como JavaScript funciona?
    - Entender o que é um compilador
    - Entender como o navegador compila o JavaScript
  - O que é JavaScript?
    - Entender o que são linguagens de programação
    - Entender como um programa de computador funciona
  - Para que serve JavaScript?
    - Entender como funciona esse trem de front-end e back-end
    - Entender como usar JavaScript no meu dia a dia

Perceba como, se não nos atentarmos ao fluxo lógico, podemos querer começar o aprendizado pela última etapa. Que, neste caso, seria a prática de construir uma aplicação inteira em JavaScript, "Construir um TODO List com JavaScript". O tempo todo estou falando sobre grade de estudos, então agora vamos utilizar essa ideia.

Para facilitar a nossa vida, podemos tanto racionalizar em cima do que temos no mapa mental, quanto procurar na internet se algum curso ou livro tem uma estrutura que possamos copiar. No nosso exemplo, vou racionalizar em cima dos tópicos, como se não tivéssemos algum recurso similar para nos basearmos.

1. Visto que eu não sei nada sobre JavaScript, o que seria mais importante entender primeiro? R: O que é JavaScript e como JavaScript funciona.

2. Uma vez que eu entendi o que é o JavaScript, como eu posso usar isso? R: Para que serve JavaScript.

3. Agora que eu sei o que é JavaScript, entendo como isso funciona e para que serve, posso seguir para o aprendizado profundo sobre a linguagem, qual o próximo tema? R: Introdução a programação com JavaScript.

4. Agora que eu sei a linguagem JavaScript, como posso profissionalizar meu conhecimento? R: Subir de nível em programação com Programação Orientada a Objetos e JavaScript e Programação Funcional com JavaScript.

Por último, só sobrou a prática mais complexa. Vale lembrar que você deve praticar em todas as arestas do nosso mapa (ou sempre que possível), afinal a prática espaçada vai fazer diferença no nosso aprendizado e, na hora de construir a aplicação inteira, vamos colocar todo o nosso conhecimento em teste de uma vez só.

Podemos, agora, quebrar o nosso mapa em mais pedaços:

- Aprender JavaScript
    - Primeiros passos
        - O que é JavaScript?
            - Entender o que são linguagens de programação
            - Entender como um programa de computador funciona
        - Como JavaScript funciona?

            - Entender o que é um compilador
            - Entender como o navegador compila o JavaScript
    - Aplicabilidade
        - Para que serve JavaScript?
            - Entender como funciona esse trem de front-end e back-end
            - Entender como usar JavaScript no meu dia a dia
        - Introdução a programação com JavaScript
    - JavaScript profissional
        - Programação Funcional com JavaScript
        - Programação Orientada a Objetos e JavaScript
    - Trabalho de conclusão
        - Construir um TODO List com JavaScript

Claro que, no meu exemplo, estou sendo bem simplista. Um mapa completo sobre aprendizagem de uma linguagem de programação, como o JavaScript, ficaria imenso. Mas não se preocupe com o tamanho do seu mapa. Ele só reflete coisas que você precisa aprender, não o tempo que vai demorar. Quanto mais você conseguir quebrar uma tarefa em menores, maior a chance de você não desanimar no caminho.

3.6 Validando o conhecimento

OK, até aqui pesquisamos sobre um assunto, construímos uma grade de estudos com dificuldade gradual e repetição espaçada de atividades práticas. Mas como eu sei se realmente estou aprendendo ou se estou pronto para enfrentar os desafios do mercado de trabalho ou de um projeto que desejo tocar?

Existem diversas maneiras de fazer isso, como: compartilhar o que aprendemos em um blog, construir projetos open source, fazer palestras ou mesmo pegar desafios de empresas e fazer eles.

Existem diversos desafios abertos na internet, pesquise por "desafio técnico de x", onde *x* é o que você deseja validar.

Compartilhar o que aprendemos, além de muito legal e útil socialmente, é muito importante para melhor associação do conteúdo e forçar nossa mente a criar abstrações para o que estudamos. Isso faz com que o assunto seja realmente fixado em nossa cabeça, pois, para ensinar, não basta repetir um texto pronto, precisamos descobrir outros modos de falar sobre aquilo do nosso próprio modo.

Explicar algo para alguém com uma linguagem diferente da que você aprendeu é ter que forçar muito seu cérebro para aprender de verdade sobre aquele tópico. Tente explicar para uma pessoa de mais idade, que não saiba nada sobre sua área de atuação, o que você faz em seu trabalho. Você vai ver como é ter que abstrair algo (além de dar algumas risadas).

Hoje a internet está aqui para facilitar a validação do nosso conhecimento. Você pode escrever um artigo na internet em um blog pessoal ou em uma publicação e aguardar os *feedbacks*. Se você conseguiu transmitir o conhecimento de maneira mais abstrata você vai receber um feedback muito positivo, senão as pessoas vão perguntar para você sobre aquilo e é aí que você vai ter que se esforçar cada vez mais para conseguir explicar o conteúdo e vai aprendendo cada vez mais.

Aprendizagem em grupo

Durante a evolução da nossa espécie, o ser humano aprendeu que criar grupos o ajudaria a perpetuar sua espécie. Você provavelmente gosta de estar em grupo ou de fazer parte de alguma coisa, mesmo que distante. Grupos de estudos podem ajudar a fortalecer ainda mais a parte de ensinar/compartilhar o que aprendeu.

Você pode formar grupos de estudos com o mesmo objetivo, discutir juntos(as) o que será estudado, buscar o máximo de informação,

montar o mapa mental, o fluxo lógico e planejar um modelo onde todas as pessoas do grupo são forçadas a “dar a aula” ao menos uma vez por assunto. Essa atividade é muito legal e outras pessoas podem ainda nos motivar no dia a dia para não desistir de continuar a nossa caminhada.

Existem diversas maneiras de se organizar um grupo de estudos, o foco aqui é: as pessoas vão conseguir fazer coisas com mais agilidade se trabalharem juntas de maneira organizada em prol do mesmo objetivo. Se quiser ir rápido, vá sozinho(a), mas se quiser ir longe, vá em grupo.

Conclusão

Até aqui chegamos a uma maneira mais autônoma de se aprender, que é a busca pelo conhecimento por conta própria. Assumir as rédeas do conhecimento vai lhe dar energia para não parar no meio do caminho. Mas, não se esqueça que nenhum método deve se tornar um molde em sua vida. Se necessário mude quantas vezes precisar até alcançar algo que funciona para a sua realidade.

Use tudo isso ao seu favor para aprender de verdade e não para decorar coisas para passar em provas e nunca mais lembrar do que foi feito.

Boa sorte em seu processo de reaprender a aprender.

Capítulo 4
Problemas ou armadilhas de sermos autodidatas

No capítulo anterior, conhecemos como foi o meu processo ao decidir aprender desenvolvimento de software e também algumas dicas para quem deseja estudar de uma maneira mais eficiente assumindo o papel de protagonista em seu aprendizado. Talvez você esteja agora querendo aprender tudo por conta própria e seguir o caminho do autodidatismo e isso pode ser bem legal.

Autodidata é a pessoa que tem a capacidade de administrar seu próprio aprendizado. Mas todo caminho possui curvas sinuosas que podem nos atrapalhar e o autodidatismo também tem. Vou lhe contar quais são e como escapar deles.

Quem assume esse papel em sua vida às vezes passa por alguns desafios que só podem ser resolvidos com o tempo, experiência de carreira ou através de conselhos de bons mentores e mentoras, como aconteceu comigo. Ou através de um livro como este aqui. Em diversos momentos eu tive dúvidas sobre se estava indo na direção certa, o que nos traz ansiedade, e precisei validar com pessoas que já passaram pelo mesmo processo para garantir que estava tudo bem e conseguir me acalmar para seguir em frente.

Ser autodidata significa que conseguimos aproveitar os vários recursos atuais e aprender sem depender de que alguém compile um conteúdo e nos explique diretamente - não que isso seja melhor do que receber conhecimento através de outra pessoa, lembre-se do que falei anteriormente sobre não existir um processo melhor que o outro, mas o que funciona para você e para mim.

Podemos quebrar a cabeça com um problema no código-fonte de um projeto que pegamos no GitHub e assim aprenderemos muito, sem que ninguém tenha nos dito nada sobre como aquilo funciona,

ou montar um móvel somente olhando como as coisas se encaixam e no fim dá tudo certo, sem sobrar nenhum parafuso.

Mas somos seres humanos, falhos, e, além disso, a vida possui altos e baixos que podem nos trazer diversos desafios ao longo dela. Desafios estes, que eu senti na prática e gostaria que você não passasse por isso: não conseguir mensurar nosso conhecimento, querer aprender tudo do dia para a noite, às vezes não conseguir sair do básico em um determinado campo de estudo, acabar com a nossa saúde mental de tanto querer estudar ou aprender mais e o problema mais chato: achar que somos gênios só porque aprendemos algo sem ajuda e com isso adquirir algo muito inconveniente para o trabalho em equipe, a soberba.

Vamos passar por cada ponto e refletir sobre eles neste capítulo.

4.1 A dificuldade de mensurar o nível de conhecimento

Quando queremos entrar em uma área do mercado de trabalho, precisamos entender em qual nível de conhecimento estamos. Isso para que tenhamos confiança de aplicar em uma vaga e não "gastar uma oportunidade". Isso é besteira, não vamos perder nada se não passar em uma entrevista de emprego, OK? Mas mesmo assim, precisamos entender nosso nível para saber onde se aplicar para não cair em um processo seletivo mais complexo do que a nossa preparação e acabar com um nível de estresse maior do que o necessário.

É bem difícil mensurar nosso conhecimento se não estamos em uma sala de aula, pois lá temos atividades, exercícios e provas finais que servem como uma validação do nosso conhecimento, além de uma

grade de estudos bem definida em tópicos que vão nos fazendo subir uma escadinha do conhecimento. Visto que o grande risco aqui é a falta de validação do que já sabemos, a solução para ele seria criar algum modelo desse tipo!

Validar nosso conhecimento em programação acaba sendo mais fácil hoje graças a algumas ferramentas como o *hackerrank.com* ou *exercism.io*, que são plataformas onde praticamos algoritmos resolvendo desafios de código criados por pessoas desenvolvedoras do mundo inteiro ou por empresas de tecnologia que visam contratação. Inclusive nós também podemos criar e adicionar nossos próprios exercícios lá para auxiliar outras pessoas.

Através das plataformas de desafios de códigos, podemos testar o que conseguimos resolver sem aquele medo de perder a oportunidade de emprego. Como os desafios estão categorizados por níveis, isso também nos ajuda a entender onde estamos naquele instante. Então use e abuse dessas ferramentas. Pratique o máximo que puder e veja seu nível de conhecimento subindo junto com seu nível na ferramenta.

Algo que também me ajudou muito foi a participação em projetos open source. Usando o GitHub, uma plataforma de compartilhamento de código-fonte, podemos procurar por *issues* (tarefas) com a categoria **good for beginners** ou **good first issue** para começar a trabalhar nelas. Uma pesquisa na plataforma retornou, em agosto de 2021, 324 projetos com issues para iniciantes. Confira no link https://github.com/topics/good-first-issue.

Ao resolver essas tarefas, a nossa confiança vai aumentar para pegar problemas mais complexos e, por consequência, elevar nosso nível técnico. Assim podemos enfim tentar uma vaga ou uma promoção (ou só validar suas habilidades mesmo).

4.2 Querer aprender tudo de uma vez

Devido à demanda das empresas e os requisitos das vagas cada dia maiores a ansiedade toma conta e acabamos achando que precisamos aprender TUDO de uma só vez.

Quando encontramos, em uma vaga para "software engineer", algo assim: "experiência de programação em alguma dessas linguagens: C, C++, C#, Java, JavaScript, Go, Python, Scala, Kotlin;" (trecho extraído de uma vaga real), entramos em desespero! São 9 linguagens de programação com propósitos, características e até paradigmas diferentes! Como vamos ter experiência em todas?

A grande verdade é que não precisamos saber tudo isso. Nem mesmo com o passar dos anos dentro da área de tecnologia eu não precisei trabalhar com tantas linguagens de programação **ao mesmo tempo**. Eu comecei a programar em 2012 e até hoje trabalhei com: PHP, JavaScript e Python. E em determinado momento eu precisei me especializar em uma linguagem, então escolhi JavaScript pelas demandas e por eu querer focar em desenvolvimento front-end.

No dia a dia acabamos precisando esbarrar em alguma outra linguagem. Eu já arranhei em Ruby, C# e Lua, mas em nenhuma delas eu precisava ser especialista ou dominar essa tecnologia, eram alterações tão simples que só precisamos do conhecimento em programação para conseguir realizar. O que algumas pessoas chamam de "saber Lógica de Programação".

Então perceba: não precisa aprender tudo o que está nas vagas de emprego hoje. Você talvez nunca vai precisar.

Agora, mesmo quando temos um foco, é muita coisa para estudar e você vai precisar se segurar. Tudo tem um ponto de partida e você precisa começar por lá. Não adianta nada querer ler vários livros e acreditar que já tem tudo o que precisa, é necessário praticar. E não adianta praticar em centenas de projetos e não adquirir a teoria,

então é necessário ler. Precisamos buscar um equilíbrio entre prática e teoria.

E também não adianta começar do ponto inicial e querer pegar atalhos durante a sua caminhada. Você realmente precisa passar por todas as etapas. Não existe curso ou livro mágico que vai transformar você em especialista ou sênior em 6 meses. Isso, além de irreal, é só marketing e vai lhe tirar a paz achando que deveria realmente ter aprendido em 6 meses o que as pessoas demoram anos.

> Evite atalhos e aproveite a caminhada. Cada erro é um novo aprendizado.

4.3 Não conseguir sair do lugar ou não sentir uma evolução

Com o tempo vamos vendo em outras pessoas um nível de conhecimento talvez maior em determinadas tecnologias. Tecnologias que achamos que também precisamos aprender ao mesmo tempo em que já temos algo em que deveríamos focar no momento (naquela grade de estudos que montamos anteriormente). Devido a não ter alguém nos "obrigando" a seguir um caminho, começamos a desviar do nosso plano de estudos devido à comparação com outras pessoas. Mesmo quando estamos em uma sala de aula acabamos caindo na armadilha de paralelizar estudos aleatórios.

Isso é extremamente comum. Vemos palestras, vídeos no YouTube, artigos em blogs especializados e parece que as pessoas estão a anos luz de distância de nós. Mas calma, jovem. Isso não só não é verdade como nem deveria ser levado em consideração. **Adquirir conhecimento não é uma competição.**

Quando chegamos a um nível de comparação tão grande, vamos entrar no fluxo de ficar procurando conteúdo sobre um tema, depois sobre outro, depois sobre mais algum. Este é o sinal de que estamos entrando em um problema chamado **paralisia por análise**. Eu entro em mais detalhes neste tópico nos próximos capítulos.

Um exemplo disso é quando queremos praticar esportes. Pesquisamos tudo sobre aquele esporte e nunca vamos de fato praticá-lo. Digamos andar de skate… Pesquisamos os modelos de shape, manobras, vemos vídeos de como fazê-las, encontramos os melhores picos para andar, entramos em grupos do Facebook e até compramos o skate, mas na hora de ir colocar o plano em prática não saímos do lugar. Voltamos a pesquisar e pesquisar e pesquisar.

Comece! O importante não é começar certo, é começar de algum lugar. Você pode ajustar a sua trilha durante a ida, não é uma direção única e inflexível.

4.4 Não cuidar da saúde física e mental

A maneira mais fácil de uma pessoa autodidata se destruir é cair em um ritmo de estudos onde esquece da própria saúde. Café, energéticos, noites sem dormir, música alta pra "focar". Nada disso vai garantir o seu sucesso.

Nosso cérebro acaba caindo em fluxos de trabalho, momentos em que mantemos um bom foco em alguma tarefa e conseguimos seguir aquilo por muito tempo. Um exemplo disso é quando estamos fazendo algo divertido e ao olharmos o relógio perdemos 2, 3 ou mais horas do dia fazendo aquela mesma coisa, como jogar um jogo, ver uma série ou sair com amigos e amigas etc.

Estudar não é diferente. Em alguns momentos estamos gostando tanto do que estamos vendo e curtindo tanto a maneira como estamos evoluindo que caímos nesse fluxo e perdemos a hora de

parar. Tudo bem isso acontecer por pouco tempo, mas quando vira um hábito, se transforma em um veneno que vai acabar por derrubar você em algum momento.

> *Fluxo* (do inglês: flow) *é um estado mental de operação em que a pessoa está totalmente imersa no que está fazendo, caraterizado por um sentimento de total envolvimento e sucesso no processo da atividade.* Fonte: Wikipedia (https://pt.wikipedia.org/wiki/Fluxo_(psicologia))

Isso pode ser prejudicial por vários fatores, como: reduzir nossa carga horária de descanso (dormir é tão importante para fixar o conhecimento quanto passar horas a fio estudando), aumentar nossa ansiedade a um nível em que acordamos mais cedo querendo ler/ver um vídeo ou fazer qualquer coisa relacionada ao conteúdo que estamos aprendendo, parar de socializar para ler artigos no celular durante um passeio.

Dormir mal vai reduzir as suas capacidades mentais (concentração, memória, atenção), pode acarretar em problemas de coração, pressão e muito mais, vai colocar você em um nível de estresse e irritabilidade incontrolável. Descansar é muito importante.

A única maneira de não cair nessa armadilha é criar rotinas saudáveis para os estudos e a saúde, com horários e dias fixos para estudar, se exercitar, alongar e dormir. Muitas vezes demonizamos as rotinas, mas elas podem nos ajudar a manter um bom ritmo de vida.

Digamos que queremos aprender uma nova linguagem de programação. Temos o costume de começar a estudar vários dias seguidos, em horários indefinidos (ou quando é possível). Mas o ideal seria especificar um horário onde, todos os dias, vamos parar para estudar e depois podemos voltar as outras tarefas da vida. Pegar este ritmo, além de não nos sobrecarregar, ensina ao nosso corpo que precisamos manter o foco na tarefa "estudos" por um tempo antes de poder voltar a nos distrair com outras coisas.

Por exemplo, podemos escolher 3 dias da semana onde vamos estudar por uma hora sem parar para olhar o celular, as redes sociais ou ver conteúdo avulso não relacionado ao tópico estudado. Isso é muito mais produtivo do que estudar 2 horas ou mais todos os dias se distraindo o tempo todo com os grupos no WhatsApp.

4.5 Achar-se melhor do que outras pessoas

Devido ao fato de que, muitas vezes, uma pessoa autodidata aprende algo mais rápido do que alguém que está em uma sala de aula por conta de horários de aulas, atraso por parte dos professores, faltar às aulas, talvez professores ruins ou qualquer outro problema desse tipo, a soberba pode tomar conta de nós.

Fora estar na frente das outras pessoas, dizer para os outros que aprendeu algo sozinho alimenta nosso ego. Todo mundo acha que somos muito inteligentes.

Nossos amigos e amigas, familiares e colegas de trabalho começam a dizer que somos muito bons/boas porque aprendemos algo sem ninguém nos ensinar. É uma verdadeira massagem para o ego.

Mas nada disso nos faz sermos melhores que ninguém. Podemos aprender rápido, aprender tudo sozinhos, mas isso não garante que somos melhores profissionais que outras pessoas que fizeram um curso ou uma faculdade, só demonstra que nós aprendemos a aprender. É uma grande vantagem que temos se pensarmos em mercado e até em economia financeira (não gastar milhares em uma faculdade nos abre possibilidades de gastar com outras coisas ou até investir o dinheiro), mas não que somos melhores que ninguém por isso.

E será mesmo que precisamos ser "melhores do que alguém"?

Abaixar a bola é difícil, requer constante análise do nosso comportamento e autocrítica, mas só de não nos colocarmos como melhores do que alguém já nos ajudará a controlar essa mania de superioridade que o ser humano tem.

Sempre que você achar que é melhor só porque aprendeu tudo por conta própria, ao invés de se colocar acima de alguém, compartilhe seu conhecimento com essa pessoa. Crie um blog ou coisa do tipo. Só assim é que realmente estamos fazendo algo útil com a nossa habilidade.

Conclusão

A capacidade de estudar por conta nos traz diversas vantagens tanto profissionalmente quanto em nossas vidas pessoais, mas também esconde armadilhas, algumas que nos dão muito prazer para o ego, outras, que podem nos trazer riscos para a saúde e problemas sociais terríveis. Não existe uma maneira perfeita de escapar disso, mas espero que a reflexão trazida neste capítulo tenha ajudado você a pensar sobre o seu estado atual (caso você esteja passando por algum dos tópicos citados).

Para irmos mais longe, precisamos ir com calma e estratégia.

Capítulo 5
Pare de procurar conteúdo e comece a praticar!

Quanto mais nos aprofundamos em um tópico mais queremos ir atrás de conhecimento sobre ele e mais conteúdo aparece. Porém, essa busca interminável por conteúdo nos coloca em um fluxo teórico que nos prende por bastante tempo, mais tempo que o necessário.

A área de computação é carregada de informação e assuntos importantes que precisamos absorver ao longo da carreira. E é impressionante que, não importa qual a área em que você escolher trabalhar, um assunto sempre puxa outro, que puxa outro e assim vai. Parece um iceberg: quando começamos a acreditar que está tudo finalizado, aparece uma base imensa que ainda precisamos enfrentar.

Com esse amontoado de temas teóricos para aprender, acaba aparecendo aquele problema comentado no capítulo anterior: não conseguimos nunca saber se estamos realmente prontos para enfrentar um desafio diferente. Seja o desafio de criar um software, arrumar um emprego ou só fazer algo diferente do que fazemos, expandir a nossa zona de conforto.

Você já se pegou fazendo um curso ou lendo um livro de determinada tecnologia e quando vai colocar a mão na massa em um exercício prático você trava? Até nos sentimos mal por ter estudado tanto e não saber nem como começar o trabalho.

No começo da minha carreira eu passei por uma situação parecida. Durante meses li diversos livros de JavaScript, desde as bases até os desafios de performance que eu nunca enfrentei. Mas quando fui trabalhar com essa linguagem, não conseguia performar o básico do que uma pessoa desenvolvedora júnior deveria conseguir.

Em alguns momentos de nossas vidas, dizer que não estamos prontos (que não sabemos o suficiente) é uma desculpa inconsciente para não se arriscar um pouco, é só **autossabotagem**. É puro medo do que as pessoas vão pensar de nós, o que pode acontecer com nosso emprego ou simplesmente receio de cometer um erro tão grande que nos traumatize.

Em outros casos, é claro, é realmente necessidade de aprender de verdade, afinal podemos não ter aprendido de verdade. Se você não sabe validar seu conhecimento, volte um capítulo e leia a seção de como mensurar o seu conhecimento.

A grande dificuldade aqui é de sair da teoria e colocar a mão na massa. E programação é isso: um misto muito equilibrado de teoria e prática. Só com teoria não conseguimos colocar um software no ar. Só com a prática não conseguimos produzir algo que dure muitos anos.

A busca pela perfeição, a comparação com outras pessoas e a ansiedade nos limitam a não conseguir alcançar nosso potencial e alavancam algo chamado **paralisia por análise**.

Paralisia por análise, resumindo e simplificando muito, é o ato de pensar demais e acabar tendo um bloqueio por isso. Caímos em uma indecisão imensa por causa da quantidade de opções que nos damos através desse monte de pesquisa, achismo e pressão e entramos em um loop de começar de novo e nunca finalizar.

Neste capítulo vamos entender como superar este bloqueio e finalmente tirar nossas ideias do papel. Existem maneiras de perceber que estamos patinando, por isso vamos aprender a identificar quando estamos pensando muito antes de agir, como superar a paralisia por análise, e como conseguir seguir a vida sem travar tanto através das metas e objetivos pessoais mais concisos.

5.1 Como saber se estou pensando muito?

É bem complicado chegar aqui dizendo que você deve parar de pensar tanto e começar a colocar as suas atividades ou metas em prática. Às vezes, precisamos sim pensar mais antes de agir, são situações e contextos que precisam ser validados antes de qualquer decisão. Sempre!

Se você está em uma situação decisória para a sua vida. Algo que nunca mais vai mudar, ou pode ser extremamente difícil, como fazer uma tatuagem com o nome da pessoa com que você se relaciona agora, claro que precisa pensar muito bem antes de agir. Mas, se o contexto é algo do tipo: qual linguagem de programação aprender ou qual carreira seguir, não faz diferença. No futuro você perceberá que pode mudar quantas vezes quiser.

O que eu quero reforçar neste capítulo é o fato de que precisamos nos tornar pessoas um pouco mais pragmáticas quando se trata do universo da programação. Principalmente no quesito estudos. As tecnologias e práticas de programação mudam rapidamente e precisamos pegar essa velocidade/este ritmo com o tempo também.

Se você navegar pelo índice TIOBE (https://www.tiobe.com/tiobe-index), vai perceber o quão rápido o mercado se atualiza. Até 2017, Java dominava o cenário das linguagens mais utilizadas, em 2021, apenas 4 anos depois, Python explodiu e passou o Java no ranquing. E agora? Será que valeu a pena focar somente em Java? Será que eu deveria ter estudado Python em 2014? A resposta é que não faz diferença, é só aprender Python ou as ferramentas que usam essa linguagem e seguir a vida. É só mais uma ferramenta dentre centenas que você aprenderá em 4 anos.

Mas você passa tanto tempo pensando em qual linguagem escolher, que o tempo passa e você não escolheu nenhuma, pois o mercado está lotado delas.

Sempre que você passar uma semana inteira para decidir algo que não implica em uma mudança para a vida toda, é hora de revisar seus pensamentos e analisar se não está investindo muita energia em um pensamento que só impede você de seguir o caminho. E às vezes não é nem "investir energia" (que é algo ativo), é só postergar mesmo (algo passivo) esperando um sinal dos céus que indique o caminho...

Quando começamos a repetir conteúdo, repetir os mesmos exemplos e exercícios e mesmo assim achamos que não estamos no momento certo para deixar de fazer isso, é este o momento em que a paralisia por análise nos pegou de jeito.

Exercícios são extremamente importantes para validar nosso conhecimento, mas se você já fez 30 vezes o mesmo, então ele não está mais validando nada. Você pode até já ter decorado o desafio e só replica com um enunciado diferente em outra plataforma.

Isso acontece muito com os requisitos das vagas de emprego… Vemos aquela lista imensa, pegamos tudo aquilo para estudar e caímos no loop da análise por medo de mandar o currículo ou se candidatar na vaga e não bater as expectativas dos recrutadores. Mas no próximo capítulo conversamos sobre requisitos de vagas para pessoas desenvolvedoras.

5.2 Como superar a paralisia por análise e começar a colocar os planos em prática?

Este problema acontece com frequência por estarmos com medo do novo, por ter hoje muita informação disponível sobre um mesmo assunto que nos confundem pela falta de e por causa do perfeccionismo. Então temos que trabalhar em cima desses pontos. Claro, a melhor maneira de conseguir escapar dessa armadilha da vida é procurar terapia, mas também podemos começar fazendo

algo antes de chegar lá, ou mesmo paralelamente. Precisamos colocar os pés no chão, tentar buscar um foco, montar pequenas metas e parar de se cobrar tanto. Foi isso que me ajudou a desparalisar e chegar até aqui.

Metas

Definir metas é uma das tarefas mais complexas do planejamento de carreira e também dos estudos, por isso, no capítulo anterior, aprendemos a criar mapas mentais para nos ajudar a quebrar nossos objetivos em peças menores. Agora vamos fazer o mesmo para nossa vida como um todo.

Pense em um mapa do tesouro (não no mapa mental), coloque um objetivo final de longo prazo, algo como 5 anos. No meu caso, eu montei um plano de me tornar sênior em 5 anos. Para chegar lá, além de demorar, será necessário passar por várias etapas, desafios que podem aumentar de complexidade com o tempo, coloque pequenos pontos nesse mapa até o sucesso que vão levar você até o objetivo final. Esses pequenos pontos são as arestas do nosso mapa mental.

As metas precisam ser pequenas, claras e alcançáveis, para que você consiga perceber sua evolução e não ficar com aquela sensação de que não está aprendendo nada, de que não está conseguindo atingir as expectativas.

Vamos refletir sobre o meu objetivo com a senioridade: primeiro, precisei conversar com diversas pessoas e coletar dados sobre o que faz uma pessoa sênior, quais as características técnicas que eu precisava alcançar e também os conhecimentos necessários para esse nível. Com isso em mãos, quebrei meu mapa mental e agora tinha algo que me ajudaria a focar.

Mas, como criar uma meta alcançável para a nossa área de trabalho, que é tão vasta?

Precisamos colocar em prática o ponto de serem claras e alcançáveis. Na minha trilha para a senioridade, cheguei no seguinte plano, por algum tempo:

- Estudar JavaScript uma hora por dia (trilha técnica)
- Ministrar uma palestra por mês (trilha comportamental)
- Pegar uma tarefa mais complexa por sprint no trabalho (trilha técnica)
- Ministrar um treinamento para novos funcionários por mês (trilha comportamental)

Como pessoa sênior, eu sabia que precisava me tornar alguém bom tecnicamente, mas que só isso não é o suficiente, então tem também as missões de comportamento. Na minha lista, tenho a tarefa (estudar, ensinar, me desafiar), um tempo em que preciso realizá-la e a quantidade de vezes que vou repetir. Elas são simples, pequenas, mensuráveis (sei o tempo que demora) e acompanháveis (se eu não executar, sei que perdi a meta do mês ou do dia).

Uma maneira de validar se está tudo bem com a trilha que você montou é solicitar ajuda de alguém com mais experiência. Pode ser alguém de um nível superior na comunidade mesmo. Está cheio de pessoas em nível sênior, especialista e mais que ajuda muito, mas estão quietinhas até alguém pedir ajuda. Espalhe por aí que você tem uma missão e precisa de uma pequena ajuda.

Zona de conforto

Muitas vezes reclamamos da nossa gestão, da liderança ou da empresa onde estamos por não trabalharmos com uma tecnologia legal, não ter um desafio diferente ou por conta da demora em conseguir uma promoção. Mas também é interessante fazer uma autorreflexão e analisar se estamos repetindo um comportamento todos os dias do mesmo modo e não saímos do lugar por patinar na zona de conforto (que é tão segura e confortável).

Por que patinamos? A maioria das vezes não percebemos, mas estamos na quentinha e controlada zona de conforto, não é mesmo? É gostoso ter controle do que fazemos, ter habilidade inquestionável sobre um assunto, mas isso não nos ajuda a evoluir. Nem como profissionais, nem como pessoas.

Sair da zona de conforto é muito difícil, pois o navio ancorado está sempre seguro e é aconchegante ficar na água calma. Mas o navio não foi feito para isso. Ele foi feito para enfrentar o mar e você também. Você não estuda para ficar só na teoria, mas para resolver problemas, inventar novas soluções em software, para criar algo novo e fantástico, então vai logo fazer isso!

Existem diversos sites com desafios de programação, diversos projetos *open source* aguardando nossa contribuição e temos muitas ideias que nunca colocamos em prática. Por que será que fazemos isso? Que tal usar o que aprendeu para dar vasão as nossas ideias ou objetivos?

Comparação

As pessoas que tendem a ser perfeccionistas costumam se comparar muito e o tempo todo com as referências da área. Em tecnologia e programação, temos muitos eventos, muita gente escrevendo artigos e fazendo vídeos para compartilhar o que sabe, o que está estudando ou com o que trabalha. Você não deve se comparar com essas pessoas. Nunca!

Nós temos vivências diferentes, a vida é maior do que o que está na internet.

Essa pessoa pode ter coisas muito importante que você não tem, como tempo e recursos financeiros, e se comparar com ela só fará com que você se sinta mal. Uma pessoa com filhos não tem o mesmo tempo para escrever um artigo, codificar para open source, palestrar etc. Alguém que mora 2/3 horas, até mais, de distância do

trabalho não tem energia para chegar em casa e gravar um vídeo. Tudo é contexto de vida.

Fora o contexto histórico. Se ela aprende mais rápido do que você, é sinal de que ela aprendeu a aprender antes de você. Ela pode ter o gene abençoado pela natureza e aprende mais rápido. Esse alguém pode ter sido melhor alimentado do que você na infância. Existem milhares de variáveis que podem ter colocado essa pessoa "na sua frente" profissionalmente e não tem como saber disso.

E a comparação não é somente com as nossas referências, mas com pessoas que estão ao nosso lado no dia a dia também.

Você precisa buscar fazer o melhor com o que você tem **agora** e continuar evoluindo gradativamente e no seu tempo. Você precisa entender que também consegue chegar a esse nível quando possuir o tempo de experiência ou recursos que os outros têm e estar em visibilidade não significa ser melhor do que ninguém. A vida não é sobre ser melhor ou igual, o ser humano é único.

Conclusão

A paralisia por análise chega em diversos momentos em nossas vidas e, dificilmente, não vamos cair nela em outro momento. Talvez, quando você estiver lendo este livro eu esteja no momento de decidir entre carreira técnica ou gestão e posso estar preso na mesma armadilha da vida. Tudo bem ser assim. Nós paramos, refletimos, revisamos os passos para superar o bloqueio e seguimos a vida.

Espero que tenha ficado claro o que você precisa fazer quando se deparar com este estado de bloqueio que a ansiedade e o medo trazem para nossa vida e que as dicas tenham sido úteis para o seu crescimento profissional e pessoal.

Vale lembrar que eu não sou profissional de psicologia. Então, se mesmo com essas dicas você não se sentir confortável e continuar

com a trava, é hora de procurar ajuda profissional para que você consiga descobrir o que está prendendo você.

Capítulo 6
Sobre os requisitos das vagas de emprego em programação

Procurar um novo ou o primeiro emprego. Quando você é estudante e precisa trabalhar ao mesmo tempo, ou quando acaba o seguro desemprego ou em qualquer outra situação agravante que vivemos, quando estamos em momentos difíceis de procura por trabalho já estamos em desespero. Para piorar a nossa situação, aparecem aquelas vagas de trabalho que parecem ter sido montadas por um robô! Um robô que entrou em vários sites sobre tecnologias, não pesquisou mais nada e nem pensou no lado humano da contratação, com isso criou uma lista de requisitos desesperadora.

Se você nunca viu uma vaga em desenvolvimento de software com uma quantidade imensa de requisitos, é porque você deve ser uma pessoa que acabou de começar a procurar uma oportunidade na área e ainda não entende o que está vendo.

Várias vezes durante a nossa carreira nos perguntamos: "Será que estou pronta(o) para o mercado de trabalho?", "Será que já sei o suficiente para encarar a entrevista naquela empresa da hora?".

Depois de encher o saco de muita gente que eu conheço na área perguntando sobre requisitos para trabalhar com programação, cheguei à seguinte conclusão:

Não existe uma lista de requisitos que todo mundo precisa saber para entrar em uma empresa ou encarar determinada função... Cada empresa pede o que precisa nas pessoas desenvolvedoras para os projetos que possuem por lá.

Porém, existem pequenas atitudes que ajudam a nos preparar e talvez diminuir esse medo dos requisitos.

6.1 Estudar

Estude muito e pratique mais ainda para que não tenha dúvidas de que você tem, sim, capacidade para encarar uma vaga de emprego.

Pense em sua evolução, não naquela lista imensa do site de vagas. Você vai se colocar no mercado com o que você possui de capacidade e, a partir do momento em que sentir que aprendeu, é hora de se candidatar.

Não importa se não tem todas as habilidades solicitadas! Entrevistando pessoas para a minha equipe eu já peguei gente que não sabia a nossa *core stack* (as ferramentas mais utilizadas), mas elas sabiam tecnologias similares e como as coisas funcionam por baixo dos panos. OK. Passaram.

Pode ser que a empresa realmente precise de alguém que saiba o que está na lista, afinal não tem como sabermos qual a real necessidade deles, mas, se puderem, vão abrir uma oportunidade para você se realmente souber o que colocou em seu currículo.

Você olhou os requisitos e sabe algo daquilo, pelo menos uns 5 itens da lista, vai pra cima!

O medo de não dar conta

E se eu passar em uma oportunidade que eu não domino 100%, que eu não cumpri as centenas de itens da lista? Aí você volta a primeira parte deste capítulo: **estude!**

Você não vai ser mandado embora se não souber uma tecnologia. Eu lhe garanto isso... Já tive pessoas na minha equipe que não sabiam React, mas estavam muito dispostas a aprender. Elas aprendem. Você aprende.

Na entrevista, exponha o que você sabe e o que você não sabe e mantenha a honestidade. Será levantado tudo o que é necessidade

técnica e, quando o trabalho começar, a equipe estará pronta para essa "fraqueza".

6.2 Trabalhar em equipe é mais do que requisitos técnicos

Sinceramente, de nada vai adiantar todo o seu conhecimento em Ciência da Computação, certificados ou cursos técnicos se você não souber algumas coisinhas essenciais para o ser humano; como trabalhar em equipe, autogestão e outras áreas de conhecimento "não técnicos".

> Tecnologia eu ensino fácil, mas a equipe não vai querer trabalhar com alguém que não respeita o time e não tem disposição de melhorar como pessoa.

Aprenda a se comunicar com pessoas das mais variadas opiniões e compartilhar a sua sem ofender alguém. O mundo não está mais chato porque não pode mais fazer piada racista, está mais justo e justiça social não é viver sem graça ou não poder compartilhar o que pensa. Existe diferença entre opinião e preconceito. Você tem todo direito de não gostar da cor lilás, mas poderia se perguntar por que faz piada colocando pessoas diferentes do padrão comercial em um nível abaixo do seu, por exemplo.

Procure ter empatia com as pessoas do seu time, ninguém tem a mesma experiência de vida que você e por isso é extremamente importante não tratar todo mundo igual. Lembre-se do primeiro capítulo deste livro, estamos falando de não ser babaca. Cada pessoa é um universo e precisamos explorar essas experiências de vida antes de falar ou fazer qualquer coisa que possa magoar.

"Mas o mundo está muito chato". Não, o mundo não está chato. As pessoas pararam de se calar sobre o que lhes afeta e por isso você recebe críticas se ultrapassar alguma barreira. É extremamente importante que as pessoas imponham os seus limites para uma boa vivência em sociedade e também em equipes.

Normalmente encontramos nos requisitos das vagas somente conhecimento técnico. Aquela velha lista de ferramentas e linguagens, mas em toda entrevista a pessoa estará avaliando o seu comportamento. Algumas empresas possuem uma etapa específica para este momento, que é o processo de "fit cultural". Nessa hora estamos avaliando se você é uma pessoa com quem gostaríamos de dividir a maior parte do nosso dia, que é o horário de trabalho.

Conclusão

Não tenha medo dos requisitos e também não deixe de se candidatar em uma oportunidade por ainda não bater todo o checklist. A honestidade e a força de vontade levam você a qualquer lugar e posição (com bastante esforço, lógico).

E se eu não passar na entrevista?

Algumas empresas costumam nos entregar um feedback construtivo do que precisamos melhorar, seja técnico ou não. Mas a maioria nos envia respostas vazias ou padrão como: "decidimos seguir com outra(o) candidata(o)". Caso não a resposta seja a padrão, evite o sentimento de culpa ou de que fez algo errado. Existem centenas de variáveis que nos impedem de seguir com alguém, desde experiência ou conhecimento até mesmo o fechamento inesperado de uma posição (quando alguém corta a verba daquela vaga).

A vida continua. Continue estudando, continue buscando, continue treinando e não desista do seu sonho!

Capítulo 7
Escolha um trabalho que você ama e nunca terá que trabalhar. Mentira!

Existe uma frase clássica que esconde várias reflexões que deveríamos fazer sobre a nossa vida profissional e a relação com o trabalho:

"Escolha um trabalho que você ame e você nunca terá que trabalhar um dia em sua vida."

Supostamente, seria uma frase de Confúcio, um pensador e filósofo chinês do Período das Primaveras e Outonos (uma era na história da China entre 722 a.C. e 481 a.C.).

Só de ler essa frase, eu já sinto calafrios.

O título do capítulo dá a parecer que eu quero arrumar uma briga com palestrantes RH, empreendedorismo ou que trabalhe motivando as pessoas para grandes empresas (como coaches), porém não é isso. Meu objetivo aqui é trazer uma reflexão sobre a nossa relação com o que fazemos profissionalmente e como este relacionamento anda bem abusivo.

7.1 A busca pelo trabalho legal

Ao mesmo tempo, ninguém recomendaria que você trabalhe com algo de que não goste. Eu mesmo já disse estas frases para muita gente:

- trabalhe com o que você gosta
- busque um emprego em uma área que você ame

- trabalhe em algo com que você se motive todos os dias para voltar no dia seguinte
- faça o que ama e nunca vai sentir que está trabalhando
- busque o que gosta e não vai sentir cansaço
- se você gosta de uma área se esforce para atuar nela
- busque especialização no que você faz para nunca precisar trocar por uma área de que você não goste

Não obstante, escrevi outro livro sobre como entrar na área de programação, O Universo da Programação. Ou seja, eu trabalho com o que eu amo e ainda incentivo que outras pessoas façam isso!

De verdade, eu acredito que você deve gostar do que faz ou todos os dias de sua vida serão um purgatório. Passamos metade (ou mais) da nossa vida no nosso emprego. Mas não quero que você se iluda achando que não vai se cansar só porque trabalha com algo que acha legal.

Se trabalho fosse um lugar somente para se divertir, ele se chamaria "parque de diversões" e não "trabalho".

O emprego dos seus sonhos pode se tornar um pesadelo do dia para a noite se o ambiente for tóxico, se a gestão lhe obrigar a trabalhar todos os dias sem folga ou mesmo se seu salário não condizer com o que deveria ser o preço justo pelo que você faz.

Um ambiente legal nos anima todos os dias, mas também nos cansa, afinal fazemos isso para nos sustentar ou realizar outros sonhos, não porque é divertido. "Trabalhar" não é somente a área, o cargo ou a função, propriamente dita, mas toda a rotina e as pessoas que a envolvem, coisas que você não necessariamente vai amar, mas que fazem parte do que se chama de trabalho.

Para refletirmos: o trabalho é uma atividade que exercemos, não é o mesmo que profissão ou inspiração de vida. Quando você faz um trabalho da escola, faculdade ou curso, é um tipo de atividade com um fim. No emprego formal, muitas vezes exercemos um trabalho não relacionado à nossa profissão de fato. No exemplo da carreira

em programação, pode acontecer, em algum momento, que você precise puxar a reunião de planejamento da sprint (a *planning*), que seria função de quem tem a profissão de *team lead* ou *scrum master*.

Ou seja, deveríamos amar a nossa profissão, a ocupação que escolhemos para exercer a maior parte do tempo das nossas vidas e não necessariamente o trabalho em si. Amar ficar 8 horas ou mais por dia desempenhando uma atividade que não necessariamente traz algo de útil para a nossa vida não faz sentido.

Caso você queira uma definição formal de o que é cada coisa quando se trata de empregos, dá uma olhada no artigo no site da Indeed, *Qual a diferença entre profissão e ocupação* (https://br.indeed.com/conselho-de-carreira/desenvolvimento-de-carreira/diferenca-profissao-ocupacao) e compare com o texto *Qual a importância de entender a diferença entre trabalho e profissão* no site da Pitágoras (https://blog.pitagoras.com.br/diferenca-trabalho-e-profissao).

7.2 O emprego dos sonhos

Trabalhar é empenhar-se em algo para concretizar um objetivo. Quando trabalhamos, queremos realizar algo, resolver um problema, completar uma missão. Às vezes, pegamos uma matéria bruta e transformamos em um produto. Outras, investimos nosso tempo e capacidades mentais criando novidades.

Nos primórdios das civilizações, as pessoas trabalhavam para construir suas habitações próximo a um lago ou rio para conseguir plantar o que comer. Hoje em dia trabalhamos para empresas. Sejam microempresas, pequenas empresas, grandes empresas ou empresas gigantes. Mas o que fazemos nem sempre é tão útil para a nossa sobrevivência e isso acaba nos desmotivando com o passar do

tempo. O nosso cérebro não dá importância para o que estamos fazendo. Até existem pessoas que investem suas capacidades em alimentação através do autocultivo, manutenção de seu terreno, mas são uma exceção.

Pensando no ponto de "mudança de vida", podemos destacar que trabalhar com programação mudou totalmente a minha vida e isso eu comento em todos os lugares, seja através de artigos, dos meus livros ou mesmo em palestras e *lives*. Mas, o que trouxe a mudança foi um momento de mercado. Eu entrei em uma área de atuação que está em destaque no mercado de trabalho. As empresas de tecnologia estão crescendo cada vez mais e precisam de profissionais capacitados para prover os melhores produtos para os clientes e assim subir o seu valor de mercado e lucro anual. Não tem nada a ver com o que eu faço de fato. Escrever código o dia inteiro não provê alimento, eu não estou plantando a minha comida, mas traz o salário no final do mês e essa é a minha motivação para trabalhar.

A maioria de nós está com o CPF atrelado a um CNPJ. E não necessariamente o seu sonho está sendo realizado através do contrato de trabalho com uma empresa. Podemos amar o que fazemos, mas quando realizamos isso todos os dias para alguém, daí a vida cobra (*the life snakes*, como eu diria nas redes sociais).

Será que quem inventou que "trabalhar com o que ama é nunca precisar trabalhar" estava dizendo: trabalhe com o que ama, para realizar os sonhos de outra pessoa que ficará com a maior parte dos lucros, e nunca trabalhará na vida? Eu acredito que não. Quando realizamos algo por nós, não temos metas e prazos exaustivos que acabam com a nossa saúde mental e física. Trabalhar com muito amor para realizar o sonho de uma pessoa no cargo de CEO me levou ao *burnout* em 2019 e essa pessoa nem sabe quem sou eu.

"A síndrome de Burnout, ou síndrome do desgaste profissional, é uma situação caracterizada por exaustão física, emocional ou mental que surge geralmente devido ao acúmulo de estresse no trabalho ou relacionado aos estudos, e que ocorre com mais frequência em profissionais que têm que lidar com pressão e responsabilidade constante, como professores ou profissionais de saúde por exemplo."

Fonte: *Tua Saúde* (https://www.tuasaude.com/sintomas-da-sindrome-de-burnout/)

Não à toa, de acordo com o ISMA - International Stress Management Association (http://www.ismabrasil.com.br), 33 milhões de pessoas brasileiras sofrem de *burnout* e, de acordo com a plataforma Blind (https://usblog.teamblind.com), 73% das pessoas que utilizam a plataforma e responderam a pesquisa está com o nível de estresse lá no alto.

Nós não deveríamos sonhar com um emprego, mas com experiências de vida, conhecimento ou qualquer outra coisa não relacionada ao velho mundo capitalista de gerar lucros para uma empresa em troca do nosso tempo de vida. Um emprego é só isso, um contrato entre você e sua empregadora.

Devo lembrar que eu não sou contra empresas, não sou contra o vínculo entre CNPJ e CPF/CNPJ e muito menos contra a frase, só quero que você não se iluda. Ao contrário do que parece até agora, eu dedico boa parte do meu tempo para motivar as pessoas a trabalharem ou buscarem emprego com o que realmente gostam de fazer. Só quero que pensemos um pouco mais em relação à frase em questão e que sejamos mais realistas quanto a **o que é trabalho** de fato.

7.3 Existem empregos legais

Até agora eu nem citei o fato de que muitas empresas sugam o máximo do funcionário. Extraem até a última gota de tempo que a pessoa poderia ter. Alguns amigos reclamam de trabalhar finais de semana, virar dias no trabalho sem voltar para casa para descansar. Eu também já passei por isso e não quero que você caia nessa armadilha.

Algumas das empresas que fazem isso com os funcionários o fazem porque têm videogame, patinete no escritório, salários altos (muitas vezes salários imensos), bons benefícios. Por isso alguns gestores se acham no direito de tirar o tempo útil da equipe em prol de bater suas metas e conquistar mais cargos dentro da organização.

Algumas pessoas já nem acham isso ruim. Acreditam que ganhar muito dinheiro e viver dentro de uma empresa seja o sinônimo de viver bem. Grandes metrópoles, como São Paulo, estão cheias dessas pessoas, empresas e gestores. Como exemplo, podemos pensar em pessoas que recebem ligação de final de semana, mensagens nos aplicativos de comunicação depois do horário e acreditam que isso é extremamente comum e até necessário. Quando estamos próximos ao *burnout*, a viagem de férias ou ida a praia é acompanhada do celular para caso alguém do trabalho precise nos contatar.

Algumas instituições precisaram mudar, pois estavam perdendo seus funcionários, outras já nasceram com uma visão de gestão totalmente diferente do que acontecia no passado. Talvez tenhamos que peneirar bastante até encontrar esse local maneiro, mas elas existem. Ou então precisemos passar por experiências ruins para enfim encontrar um ambiente ok.

Na nossa área temos até muita sorte. Com o avanço dos métodos ágeis, popularizados em 2001, temos a pessoa no centro do processo de desenvolvimento ao invés de meios engessados de gestão que só traziam estresse e mais tensão para o nosso dia a dia.

Mas, então, como encontrar um bom lugar? Como em todas as minhas palestras, mais uma vez eu vou lhe pedir para se apoiar nas comunidades de programação para conquistar um emprego legal de verdade. Além de não se contentar em ficar em um lugar ruim, podemos contar com a ajuda das comunidades para saber se um emprego é realmente bom.

Existem plataformas, como o Glassdoor (https://www.glassdoor.com.br) que colocam empresas em rankings onde nós damos as notas por benefícios, cultura, ambiente de trabalho etc. Isso pode nos ajudar na pesquisa por uma boa organização, mas os dados são facilmente manipulados por quem tem muito dinheiro e não podemos acreditar somente nessas plataformas. Encontre uma empresa com uma boa pontuação e vá atrás de quem trabalha por lá nas comunidades de programação. Nós sempre compartilhamos nossa opinião sincera sobre o local.

Conclusão

Escrevi este capítulo porque muita gente vem até mim perguntando se é divertido trabalhar como desenvolvedor de software e citam essa frase, então eu sempre respondo tudo isso que disse até aqui. Trabalhar como desenvolvedor é uma coisa muito divertida, legal. Só que nem todos os dias são flores, é trabalho.

Qualquer profissão tem altos e baixos. Momentos onde estamos desempenhando o melhor em tarefas legais, que transformam a vida das pessoas, causa um impacto positivo ao nosso redor, mas também aqueles onde precisamos preencher uma planilha chata ou fazer algo totalmente rotineiro. Um show de dança é emocionante para quem está apresentando, mas os ensaios e treinos físicos, muitas vezes, são feitos na base da persistência, pois é difícil encontrar motivação para essa parte "chata" dessa profissão.

Eu acredito, sim, que dá para trabalhar para uma empresa e ser feliz. Colegas legais, processos que não nos limitam, ambiente acolhedor que extrai o nosso melhor... Isso existe. Mas, dizer que não está trabalhando é uma grande mentira capitalista para que não sejamos inconformados com os problemas sistêmicos das organizações.

Busque fazer o que gosta, mas não se iluda achando que todos os seus dias serão maravilhosos. Seja realista consigo para que não sofra quando os dias ruins baterem na porta. Também permita-se ter dias ruins, aqueles onde você não terá capacidade para produzir tanto e não vai conseguir se empenhar. Isso faz parte do nosso dia a dia.

Espero que este capítulo tenha clareado um pouco suas ideias e preparado você para quando ouvir essa frase de algum entrevistador. Mas que também tenha lhe dado esperança de um dia trabalhar em uma empresa legal de verdade onde realmente seu trabalho seja divertido e produtivo, sem que você perca seus finais de semana e momentos de cultura e lazer.

Capítulo 8
Quando abandonar um emprego

Sair de um emprego é uma decisão muito complexa. Nosso corpo entra em um nível de estresse e ansiedade imenso tentando computar todas as possíveis variáveis da troca.

Temos que pensar em coisas como: salário, ambiente, cultura, evolução profissional, benefícios, regime contratual, o teste prático, a entrevista e talvez mais.

A indecisão é ainda maior quando estamos com grandes responsabilidades, como filhos, aluguel e similares. O medo de trocar de organização, não dar certo e perder o emprego é gritante em nossa cabeça.

E nós temos mesmo que pensar em tudo isso quando estamos em um bom lugar e decidimos sair dele por motivos comuns. "Comuns", que eu digo, seriam: o lugar ser muito longe da nossa casa, os benefícios são ruins, o salário está defasado, precisamos de desafios maiores. Esses pontos não categorizam um emprego como ruim, porque, se o emprego for bom mesmo, podemos negociar todas essas coisas.

Neste capítulo vamos refletir um pouco sobre o que é um ambiente de trabalho tóxico, sobre a nossa esperança de lugares ruins mudarem da água para o vinho e como pequenas escolhas impactam grandemente o nosso futuro profissional e pessoal.

8.1 O ambiente tóxico

Temos diversas visões sobre o que é um emprego ruim. Aquele que paga salário ruim ou onde temos que trabalhar de finais de semana.

Isso varia de pessoa para pessoa. Em minha visão esses detalhes são pormenores que conseguimos superar com o tempo, capacitação, trocando de equipes ou empresas. Mas algo que pode nos afetar tão profundamente ao ponto de atrapalhar as nossas vidas com traumas é o ambiente tóxico.

É quando você não sente vontade de levantar da cama para ir para o trabalho porque a liderança da equipe é uma pessoa difícil de lidar, que não respeita você ou não entende as suas individualidades. Quando o ambiente é tão machista, preconceituoso, racista, homofóbico, que só quem se sente bem lá são os homens brancos, barbudos e bombados de classe média alta da nossa área (chamo-os de "time BBB").

Se você está nesse tipo de ambiente, é hora de começar a refletir se vale mais a pena abandonar esse barco ou continuar arriscando sua sanidade mental. O ambiente está ruim dessa forma porque a cultura da empresa é ou está ruim. E, na maioria das vezes, isso não vai mudar. Por mais que nós relatemos os problemas para a gestão ou para o RH. Eles simplesmente não querem, não podem ou não conseguem quebrar esse ciclo.

Isso mesmo, a maioria das vezes não é culpa da sua gestão ou do time de RH, talvez nem da pessoa presidente da sua empresa. Gerenciar uma empresa é muito maior do que imaginamos. De vez em quando minimizamos um problema achando que é culpa da pessoa X do RH, mas essa pode estar arrancando os cabelos tentando mudar a cultura da empresa e isso nem chega ao nosso conhecimento.

Quem nunca reclamou de um companheiro de trabalho arrogante e a liderança não fez nada, que atire a primeira pedra. Se as pessoas não respeitam nossas limitações, se não nos tratam com igualdade na hora de transmitir conhecimento, não têm empatia conosco é porque quem as contratou errou na hora do *fit cultural* ou a cultura é ruim mesmo e eles não ligam.

Perceba: fit cultural é o alinhamento ideológico entre a pessoa que vai entrar com a cultura da empresa. Se a pessoa que está ao seu lado é alguém com péssimo comportamento, ou a empresa errou na contratação (acontece muito) ou é um péssimo lugar para se estar.

Olhe ao redor e comece a analisar: o que você vê são mais pessoas em um padrão negativo ou poucas pessoas se destacam por fazer algo não tão legal?

Se a resposta for que temos muitas pessoas com comportamento ruim, é um mau sinal.

8.2 Uma empresa ruim pode nunca deixar de ser

As pessoas mudam, o ambiente de trabalho pode, sim, melhorar com a entrada de pessoas legais. Alguns amigos discordam dessa minha "fé nas pessoas", mas eu já vi ambientes aos quais eu não gostaria de voltar se transformando em outra coisa e depois eu até repensei sobre o tema. Mas isso não é uma regra. Na realidade a mudança de uma cultura organizacional é uma grande exceção.

No mundo dos games tivemos uma grande aquisição em 2022. A Microsoft, empresa que já teve e ainda deve ter vários problemas de cultura, mas está trabalhando nisso, comprou a Activision Blizzard, outra empresa que está passando por péssimos casos de abuso de poder até mesmo abuso sexual. Com essa compra, o mercado acredita que uma mudança de cultura acontecerá, os casos de assédio vão reduzir ou até acabar, mas é uma compra em milhões de empresas com milhares de pessoas trabalhando. Pode ser que isso aconteça no local em que você está, mas pode não acontecer. Quem precisa lidar com isso é você, hoje.

Mas essa minha esperança, que também pode ser a sua, pode nos prender a uma âncora que nos leva direto aos problemas de saúde causados pelo estresse. Nós acreditamos na mudança e, nessa esperança, continuamos no lugar. Só que isso pode nunca acontecer. Os pontos negativos do lugar podem ser coisas que as pessoas já estão acostumadas e não vão fazer nada para transformar em melhorias.

Eu acredito que conseguimos mudar um ambiente através da comunicação, melhoria de processos e a descentralização de poderes. Porém, temos que manter os pés no chão e observar se nossas atitudes estão gerando frutos no local ou se só estamos sonhando alto enquanto mergulhamos fundo no oceano do estresse.

Se as pessoas ao redor estão questionando e a empresa nada muda, é sinal de que nada vai acontecer. É hora de sair desse emprego.

8.3 O impacto das nossas escolhas no nosso futuro

Eu não vou jogar a culpa dos problemas em suas escolhas. Eu não acredito em culpa. As coisas acontecem por milhares de variáveis que acabam trazendo você para onde está hoje. Se dependesse somente da sua escolha, tenho certeza de que você buscaria sempre o melhor para a sua vida, não é mesmo?

Mas, quando estamos sentindo que o ambiente não está legal e estamos começando a tomar prejuízos, seja na mente ou no corpo, é hora de tomar uma escolha para a vida. Nesse momento, é importante buscar as melhores opções. As pessoas chegam a momentos ruins em suas vidas por conta de não terem boas opções para escolher.

Reflita comigo: se eu tenho a opção de assistir uma série que detesto ou um filme que eu não gosto, não importa a minha escolha, eu vou me dar mal.

Nesse momento, o autoconhecimento será extremamente necessário. O acompanhamento psicológico foi a melhor coisa que me aconteceu nos momentos ruins em que o *burnout* estava me pegando. Consegui entender o que me fazia bem e o que me fazia mal no ambiente de trabalho, aprendi a impor limites sem precisar me exaltar, e isso foi maravilhoso.

Quando estiver no momento em que não está se sentindo muito bem e perceber que isso é devido ao trabalho, será necessário buscar melhorias. Seja através do acompanhamento terapêutico ou mesmo fazer um curso, estudar, treinar, para procurar um novo emprego.

Não espere o seu pavio chegar ao fim para buscar ajuda, pode ser muito perigoso.

Conclusão

Este capítulo não é direcionado a nenhuma empresa em especial, mas uma reflexão pessoal que carrego comigo depois de ouvir vários relatos de empregos ruins que as pessoas me mandam durante o processo de mentoria que eu presto para guiar a carreira delas desde 2014.

Se você está em um ambiente ruim, como os pontos que citei, procure sair logo daí! Não temos poder de mudança da parte de baixo da pirâmide hierárquica e nosso tempo é bem valioso para que percamos ele por amor a um CNPJ.

"Nenhum CNPJ vale um AVC" - Flávia Gamonar, TEDxUFTM.

Nossa saúde não é como um *code base* com controle de versão. Não podemos reverter um problema psicológico facilmente. Se nos deixarmos levar pela esperança de mudança em troca de perder

nossa sanidade, teremos pago um preço alto demais por uma possível ilusão.

A relação empresa e funcionário(a) deve ser de troca mútua: nós damos lucros e a empresa nos proporciona um ambiente saudável. Você não está no emprego por favor da empresa, você está ali porque seu trabalho gera lucro. Se você sair por não conseguir mais trabalhar, eles simplesmente trocam você por outra pessoa, então porque não trocar de empresa quando ela é que é a ponta ruim?

Capítulo 9
Devolva para as comunidades

Todos os dias usamos as comunidades de programação e até mesmo tecnologias que as comunidades criam e, sem perceber, não retribuímos por isso. Nosso saldo com as comunidades de programação está no negativo e eu posso provar para você!

A qualquer momento do nosso dia utilizamos softwares *open source*. Não importa o quê e nem como você faz, código aberto e/ou software livre sempre estão ali nos programas que você utiliza, como os editores de textos VIM, Atom ou VS Code, Firefox ou o fork do Chromium, o Google Chrome ou, aprofundando mais ainda, a maioria dos sites acessados ou aplicativos usados que estão desenvolvidos em linguagens open source, como PHP, Python e mais.

Além do uso das ferramentas, nós, como pessoas desenvolvedoras de software buscamos respostas em fóruns abertos pela internet e/ou no GitHub, como o freeCodeCamp ou Stackoverflow que são basicamente alimentados pelas comunidades. Respostas, artigos, vídeos e muito conhecimento compartilhado por pessoas que doam seu tempo em troca do bem-estar coletivo.

Onde tem open source, tem comunidade. Onde tem apoio para ajudar a tirar suas dúvidas, tem comunidade. São muitos os casos em que as comunidades nos ajudam, mas será que estamos "devolvendo" essa ajuda de alguma maneira ou estamos deixando passar a oportunidade de retribuir o trabalho dessas pessoas? Além de reconhecer o trabalho de quem contribui, será que estamos ajudando esses grupos a continuarem existindo?

A maioria das pessoas que escrevem artigos na internet, fazem vídeos, palestras e escrevem código em repositórios abertos não

recebe nada por isso. Ao contrário, alguns acabam recebendo ataques de empresas e outras pessoas da comunidade.

Em 2016, um desenvolvedor cansou de pedir ajuda para o GitHub e NPM, onde seus pacotes estavam hospedados e decidiu remover, depois da ameaça de processo por uma empresa que queria publicar um pacote com o mesmo nome de algo que ele já havia publicado. Ele decidiu remover os seus pacotes do ar e assim **quebrou a internet**.

No texto *How one developer just broke Node, Babel and thousands of projects in 11 lines of JavaScript* (https://www.theregister.com/2016/03/23/npm_left_pad_chaos/), no The Register, conseguimos entender um pouco mais sobre o que houve, mas o fato é que uma de suas bibliotecas era usada por **milhares** de sistemas que pararam de "buildar" e não podiam mais receber patches de atualizações. Azer Koçulu possuía mais de 270 pacotes publicados. Agora imagina o tanto de tempo que essa pessoa empenhou para que os sistemas que utilizamos estejam no ar e não recebeu nada por isso.

Pessoas desenvolvedoras open source estão por aí precisando do nosso apoio para continuar trabalhando por nós. Vamos refletir um pouco e aprender como apoiar as comunidades para que esse ecossistema nunca deixe de existir.

9.1 Comunidades são muito legais

Comunidades são espaços virtuais ou físicos onde pessoas se juntam para realizar algo em grupo. Podem ser espaços de troca de conhecimento, prática de desenvolvimento de software ou mesmo só para descontrair com pessoas da mesma profissão. Eu amo esses grupos. Gosto de tudo que envolve trabalhar com pessoas e mudar a vida delas de algum modo, utilizando tecnologia e programação

então? Acho ótimo! E quando penso em comunidade é transformação que me vem à mente.

Desde os primórdios do meu encontro com tecnologia da informação eu conheço, uso e abuso do que esses espaços podem nos proporcionar. No começo, lá para 2007, recebi ajuda das pessoas na internet para me envolver com manutenção de computadores, depois para aprender sobre o sistema operacional Kurumin, mais à frente sobre Ubuntu, em seguida comecei a aprender PHP, Python, JavaScript e, em todos os casos, os coletivos estavam lá para me ajudar. É sério, muita gente me ajudou a resolver problemas de impressoras e isso me dava uma graninha extra no final do mês até o momento em que eu pude parar de trabalhar com o que eu não gostava e só focar em minha carreira como programador.

Quando decidi trabalhar como desenvolvedor de software focado em front-end, busquei ajuda na internet e achei diversos guias, artigos, vídeos, tutoriais completos sobre como entrar na área. Eu já tinha aquela experiência procurando conteúdo e aprendendo com o apoio das pessoas que geram conteúdo para a web, então foi muito fácil encontrar o que eu queria devido ao grande volume de blogs e comunidades.

Em seguida fui atrás de ainda mais apoio na comunidade Front End Brasil a @frontendbr no GitHub. Mas muito antes disso eu já havia buscado ajuda em outro grupo, o PHP Brasil no Facebook, sobre como me tornar desenvolvedor de software. Percebe um padrão? Sempre que eu preciso de apoio, sei exatamente onde procurar: nos coletivos na internet.

E isso não aconteceu somente no começo da minha carreira. Hoje, como desenvolvedor sênior ou especialista em front-end, também busco as comunidades quando preciso de algo e também estou sempre lá para tirar dúvidas das pessoas iniciantes. Não é difícil me encontrar respondendo algo nos fóruns ou tópicos nas redes sociais.

9.2 Como tentamos pagar nossa dívida

Hoje tento retribuir o que eu absorvi através de várias iniciativas como os grupos de estudos, mentoria, compartilhar minha maneira de trabalhar, gerar cursos dentro do meu blog woliveiras.com.br ou fazendo lives para o meu canal na Twitch.tv (https://twitch.tv/1ilhas).

Tento compartilhar o que acho útil para outras pessoas e incentivar quem tem iniciativas abertas e, mesmo assim, ainda me sinto em dívida com quem me ajudou e ajuda todos os dias de alguma maneira, pois foi apoio demais.

Talvez, com o começo desta reflexão, você tenha se animado um pouco com o que as comunidades fazem e queira contribuir, então como fazer isso (ainda mais se for iniciante)?

Posso começar listando algumas coisas:

- escrevendo para blogs/portais
- respondendo em fóruns
- através de código-fonte em projetos abertos
- compartilhando o que você sabe no seu próprio blog ou redes sociais
- divulgando o trabalho das pessoas criadoras de conteúdo
- apoiando financeiramente as iniciativas abertas

Sabe aquele post que ajudou você a solucionar um problema difícil, seja nos estudos ou no trabalho? Compartilhe-o. Você estará retribuindo e incentivando a pessoa que escreveu. E ainda pode ajudar o texto a chegar até outra pessoa também está passando pelo mesmo problema. Comente o quanto o texto ajudou naquela caixinha de comentários ou marca essa pessoa nas redes sociais agradecendo pelo seu apoio. Acredite: quem escreveu gostaria de receber feedbacks. É muito motivador saber que o que fazemos está sendo útil para alguém.

Você achou algo errado em um artigo/vídeo? Comente sobre o erro e dê alguma dica de como melhorar o texto ou como corrigir o problema encontrado. Muitas vezes nós, que estamos de fora da ideia, conseguimos pegar pequenos problemas que a pessoa que gerou o conteúdo não consegue enxergar, por mais que revise várias vezes.

Uma correção ortográfica já ajuda muito, pois não temos revisores profissionais cuidando do nosso trabalho nos blogs! O meu blog mesmo é open source e você pode corrigir sempre que necessário ou até mesmo colocar mais exemplos de código caso acredite que faltou. Esse apoio não só me incentiva quanto impacta positivamente todas as pessoas que lerem o conteúdo depois da sua colaboração.

Sua participação ativa nos fóruns será excelente. Foram respostas cheias de conteúdo legal e boas argumentações que me ajudaram a aprender o que sei hoje em dia. Você não tem muita experiência? Não tem problema. Talvez você consiga ajudar mesmo assim. A pessoa pode estar perdida por **não saber como pesquisar sobre o assunto**, se você conseguir achar uma resposta legal no Google, compartilhe com ela!

Sabe aquele artigo que ajudou você a aprender um assunto? Que tal compartilhar com os grupos/comunidades por aí e dizer como ele lhe foi útil? Compartilhe vagas ou dicas nos fóruns. Os grupos estão lotados de gente procurando emprego, mentoria ou apoio do tipo. A maneira como eu mais aprendi foi quando eu comecei a compartilhar no blog e participar de forma ativa nas comunidades. As dúvidas de outras pessoas nos ajudam a refletir diferente sobre um mesmo tema e isso transforma o nosso conhecimento.

Tudo isso também se aplica aos projetos open source. Você não precisa ser uma ninja do código, ir ao repositório, modificar tudo ou resolver todas as *issues* abertas. Compartilhar esse repositório pelas comunidades, contribuir para melhorar a documentação do projeto e levantar ideias para possíveis melhorias é uma grande ajuda! Tem alguém com experiência e capacidade para resolver o problema em

algum lugar por aí. Talvez você consiga conectar esta pessoa ao problema que encontrou, abra a issue e compartilhe nos grupos. O impacto é enorme.

Perceba que não é código que impacta as comunidades de programação, por mais que elas se chamem comunidades **de programação**. Comunidades são sobre pessoas. Pessoas movem o mundo e fazem as comunidades valer a pena.

Ajude as pessoas pelas pessoas, não por fama

Apesar da maravilha que existe em ajudar, um alerta tem que ser feito: ajude pelas pessoas que você impacta, não por fama!

É comum que, conforme você contribua mais e mais, uma certa fama seja criada em torno do seu nome nas redes sociais. Você terá uma imagem, as pessoas vão lembrar de você e até podem reconhecê-la(o) em eventos de tecnologia.

Mas ajude pessoas sem pretensão de ganhar fama nas costas das comunidades. Não queira sair por cima com projetos bonitinhos ou só contribuir naqueles cheios de estrelas no GitHub (os mais badalados). Ajude de coração.

Tem projetos simples que são excelentes locais para você aprender alguma coisa sem pretensão de virar o próximo superpalestrante famosão das galáxias. Antes de ajudar alguém, aprenda a controlar o ego, pois ele pode crescer — somos humanos e sujeitos a falhas — e aprenda a usar sua empatia. Coloque-se no lugar de quem precisa e faça por eles. Pessoas são incríveis e você pode aprender muito com elas.

No final você sempre vai ganhar algo, seja um agradecimento, mais motivação para continuar ajudando ou aprender algo legal. Se você ficar "famoso" por isso, é só consequência do seu trabalho com os coletivos.

Conclusão

Se você não conhecia comunidades antes desta leitura, acredito que veio pelo caminho certo. Eu tenho meus problemas com as comunidades, afinal de contas não existe ambiente ideal quando junta um monte de gente pensando diferente em um lugar, alguma treta vai rolar. Mas eu amo estes espaços e espero que você receba as nossas boas-vindas quando puder contribuir.

Não precisa ser especialista para começar a ajudar. Ajude com o que você puder e se puder. Não é para transformar o apoio as comunidades um peso em sua vida, precisa ser algo legal de se fazer. Aproveite a viagem.

Capítulo 10
Por que existem cursos de programação exclusivos para LGBTs e outras minorias?

Algum tempo atrás compartilhei um curso de programação exclusivo para **pessoas trans** em um grupo de programação no WhatsApp. O curso em questão era o da *transcendemos*, conteúdo gratuito e exclusivo para pessoas trans e travestis. Como era de se esperar, logo alguém me perguntou: "Hein!? Por que um curso exclusivo para travestis, essas pessoas não podem simplesmente entrar em qualquer curso de programação, **como eu ou qualquer um aqui no grupo?**".

Alguns exemplos de cursos de programação exclusivos para alguns grupos sociais, para exemplificar:

- Mulheres: Reprograma, PrograMaria, Laboratória
- Pessoas de baixa renda: Tecnogueto, Núcleo de tecnologia do MTST
- Pessoas negras: Código Preto
- Pessoas com deficiência: Web para Todos
- Pessoas LGBT e outros grupos: Juntos na TI

São muitos grupos e diversos alvos de inclusão e, independente do público-alvo, essa pergunta sempre aparece. Estou acostumado com isso, afinal trabalho com diversidade e inclusão desde 2015 e já vi essa pergunta se repetir mais vezes do que temos vontade de responder. Isso não é nenhum pecado, você não é uma má pessoa por questionar sobre esse tema. A maioria das pessoas se pergunta o mesmo, só não tem coragem de expor o que pensa. Então pensei em sintetizar o que sei sobre o assunto neste capítulo para facilitar o nosso entendimento e, quem sabe, conseguir esclarecer sobre por que existem cursos exclusivos para minorias sociais, não somente para pessoas trans, mas também para mulheres, pessoas negras e afins.

Para começar, vamos refletir sobre o contexto das pessoas em situação de vulnerabilidade social, depois passar sobre representatividade e, enfim, concluir nosso raciocínio sobre inclusão. Caso ainda tenha alguma questão sobre isso, pode me chamar nas redes sociais (@1ilhas no Twitter e @1ilhas na Twitch), estou à disposição para esclarecer suas dúvidas.

10.1 O ambiente social é excludente

Para começar a entender o porquê dos cursos exclusivos para pessoas trans, mulheres, pessoas negras, indígenas, PCDs e outros grupos sub-representados, precisamos observar que a sociedade é por si excludente. A nossa criação é cheia de exemplos do que não deveríamos carregar para as nossas vidas, mas levamos para a frente por ser normalizado.

Por um momento na nossa história ser uma pessoa branca, europeia, era o ser correto. Ter nascido uma pessoa negra significava existir para servir as pessoas brancas. Até mesmo entre os brancos, tivemos o caso de pessoas que acreditaram na "raça ariana" que os colocava acima de tudo e todos. A igreja, inclusive, categorizava pessoas negras como seres sem alma. A mulher também sempre foi colocada em posição de servidão, uma vez que o símbolo de padrão era o homem, hétero e "bem-sucedido".

Entenda, existem vários tipos de minorias e isso vai depender também do país onde vivemos e do momento histórico em que vivemos. Segundo o portal Brasil Escola, temos a seguinte definição:

O conceito de minoria social diz respeito, nas ciências sociais, a uma parcela da população que se encontra, de algum modo, marginalizada, ou seja, excluída do processo de socialização. São grupos que, em geral, são compostos por um número grande de pessoas (na maioria das vezes, são a maioria absoluta em números), mas que são excluídos por questões relativas à classe social, ao gênero, à orientação sexual, à origem étnica, ao porte de necessidades especiais, entre outras razões.

Fonte: Brasil Escola - Minorias sociais (https://brasilescola.uol.com.br/sociologia/minorias-sociais.htm)

Não precisa ser algo tão "diferente", somente o fato de ser uma pessoa periférica, que se veste como acha legal dentro de um ambiente onde o padrão é seguir um estilo mais classe média alta, distante da cultura hip-hop, por exemplo, você já vai sofrer algum tipo de preconceito e microagressões.

Veja o caso do Tiago, hoje conhecido como O Chavoso da USP, um jovem periférico que sofreu preconceito pelo modo como se veste, remetendo a cultura funk atual, e falar utilizando girias comuns da periferia em um ambiente onde o padrão é a norma "culta" e a cultura periférica é vista como marginal, a universidade. No portal UOL você encontra uma matéria sobre o rapaz, procure por "Thiago Torres o Chavoso da USP", ou assista o seu canal no YouTube, onde ele conta sua vivência como pessoa periférica dentro de um ambiente excludente.

Na minha vivência, como alguém que se vestia basicamente no modelo hip-hop, muita gente da área de programação zombava de mim por eu me vestir "como um maloqueiro". Perceba: as microagressões são diárias e pode ser algo pequeno ou mesmo um atentado a sua vida. Pessoas trans, visto o exemplo do curso citado, correm risco de vida todos os dias somente por existir.

Microagressões são pequenos atos de preconceito que acabam infringindo grandes danos à saúde mental e até mesmo à integridade física das pessoas. É comum observar senhoras ou senhores de mais idade trocando de calçada ao ver uma pessoa negra vindo em sua direção, também acontece o mesmo quando veem uma pessoa trans ou um casal homossexual andando de mãos dadas. Imagina o quanto isso é triste para a outra pessoa.

Essa violência, muitas vezes velada, acaba excluindo as pessoas de grupos sub-representados pouco a pouco até elas não estarem mais em ambientes que deveriam ser de comum acesso a toda a população. Não é à toa que os cursos de tecnologia nas maiores universidades são frequentados majoritariamente por homens brancos de classe média alta. Com o tempo, as pessoas que não fazem parte desse padrão começam a se sentir mal por não encontrar outros iguais, são excluídas e acabam evadindo da graduação (ou do curso técnico ou até mesmo da formação básica escolar).

> Na década de 1970, cerca de 70% dos alunos do curso de Ciências da Computação, no IME, eram mulheres; hoje, 15%.
>
> Fonte: Jornal da USP - Por que as mulheres "desapareceram" dos cursos de computação? (https://jornal.usp.br/universidade/por-que-as-mulheres-desapareceram-dos-cursos-de-computacao/)

Nós somos seres sociais e todo mundo tem a necessidade de ser aceito no ambiente onde está, para que se sinta bem o suficiente para continuar fazendo o que faz. Seja em um curso, no trabalho, na família ou no círculo de amizades. É por isso que você, assim como eu, vive em uma bolha, procurando grupos de programação onde possa falar sobre isso e não sobre outros assuntos. Se você gosta de futebol, se aproxima de quem gosta de futebol, se curte cinema, se aproxima de quem gosta de cinema, e assim vai.

Porém, não é o mesmo com representatividade. Não é só gostar de algo, como uma roupa, é sobre **ser** "diferente". Ser uma pessoa trans ou travesti, ser gay, lésbica ou bissexual nos coloca um peso de estar em lugares onde você pode receber julgamento por qualquer atitude ou fala. Só pelo fato de você existir ou amar alguém. Se você for uma pessoa heterossexual, pode refletir que isso nunca foi um peso em uma entrevista de emprego. Você nunca teve medo de falar “eu sou hétero” e perder a oportunidade de trabalho.

Em um ambiente como um curso não é diferente. As pessoas estão ao redor julgando o tempo todo. Não é por maldade, às vezes é uma brincadeira que ela costuma repetir, por outras vezes é algo que os pais falam e a pessoa replica, e isso se torna exaustivo para quem é o alvo dessas microagressões. Quando você faz parte de uma minoria social, já não basta o peso que é correr atrás de uma vida profissional estável, estudos e tudo mais, ainda existe um agravante de simplesmente ser quem você é.

É por essa exclusão natural do convívio social que existe a necessidade de criar grupos onde as pessoas se sintam representadas e que não sejam julgadas ou questionadas por serem como são. Seja mulher, LGBT, negras ou qualquer grupo. Assim, a evasão será reduzida e o impacto social do curso será muito maior.

10.2 Não é sobre você

Toda vez que alguém se sente desconfortável com cursos de programação ou vagas de emprego com exclusividade para minorias sociais, ela o faz em questionamento sobre a necessidade de que todo mundo siga o mesmo caminho - o caminho que essa pessoa trilhou - desconsiderando totalmente as vivências individuais de cada um.

Se eu participei de um curso não exclusivo só pra mim, por que essa pessoa trans, negra, tem esse **PRIVILÉGIO**?

O ponto é que a exclusividade do curso não é sobre você, meu amigo ou minha amiga. A sua bolha social pode ter algumas vantagens que você nunca parou para reparar, assim como eu, um homem, nunca havia parado para entender porque não tinham mulheres trabalhando comigo até começar a participar ativamente das comunidades de programação.

A necessidade da exclusividade é pela representatividade, nada mais. A pessoa que está em um grupo que a aceita vai se sentir mais à vontade para tirar dúvidas, interagir com a turma, ir à casa de alguém para estudar. Ninguém quer sair na sua frente, só quer se colocar ao seu lado, nas mesmas oportunidades de verdade, pois quem está à frente no mercado é quem não faz parte de um grupo marginalizado.

Como homens, nunca tive esse problema. Bastava falar sobre futebol e todo mundo virava amigo. Quando se formam grupos, o Clube do Bolinha está formado e nós excluímos as mulheres sem nem perceber. É triste, mas é uma realidade que nós muitas vezes não reparamos.

Olhe ao seu redor no emprego onde você está e reflita: quantas mulheres tem em sua equipe? Quantas pessoas negras estão em cargos de liderança? E quantas pessoas trans estão na diretoria da sua empresa? Quantos LGBTQ+ são assumidos na sua profissão?

Cursos exclusivos são sobre necessidades sociais, não é sobre tecnologia, não é sobre tomar o seu lugar e nem é uma competição.

Existe um vídeo legal que pode lhe interessar, se você quiser entender mais sobre privilégios. Chama-se: a corrida dos privilégios. Você consegue encontrá-lo no YouTube e isso vai ajudar você a refletir bastante por alguns minutos.

Conclusão

Espero que, com essas palavras, tenha ficado claro que o grande ponto aqui é a representatividade e a socialização. Ninguém gosta de estar onde não é bem-vindo(a) e, pior ainda, quando é algo que nos traz ansiedade, como pensar em futuro profissional.

Daqui para a frente, quando você descobrir um curso exclusivo para minorias, ao invés de questionar, torço de coração para que você compartilhe sobre ele com o máximo de pessoas possível para que as diferenças sociais comecem a diminuir e no futuro não seja mais necessário que existam iniciativas exclusivas, mas que as pessoas sejam bem-vindas onde quer que seja.

- "Quero que a periferia possa seguir trajetória que quiser" (https://www.uol.com.br/ecoa/ultimas-noticias/2020/09/28/chavoso-da-usp-quero-que-a-periferia-possa-seguir-trajetoria-que-quiser.htm).
- Perfil do Tiago no Instagram: https://www.instagram.com/chavosodausp
- Canal do Tiago no YouTube: https://www.youtube.com/c/ChavosodaUSP

Capítulo 11
A vaidade em engenharia de software

Todo ser humano, mais cedo ou mais tarde, vai cair em uma furada por conta da vaidade. É uma característica nossa e é até condenada por algumas religiões, de tão perigoso que é cair nesse mal. Ser uma pessoa vaidosa não é de um todo ruim, cuidar da nossa aparência, por exemplo, não é algo que nos fará mal. O ruim de qualquer característica humana que pode causar problemas é o excesso. E, por excesso de vaidade, já vi muitas pessoas desenvolvedoras de software caindo em empregos e projetos ruins.

Para deixar claro do que estou falando neste capítulo, quando digo vaidade, me refiro ao comportamento de colocarmos valor em algo que não realmente tem um valor prático. É basicamente uma ilusão que pode ter chegado até nós através do contexto social ou até mesmo pelo incentivo da comunidade de programação para algumas atitudes.

Na nossa área, a vaidade vai nos fazer trocar de emprego sem necessidade, aceitar um salário ruim por causa de um título, entrar em uma empresa ruim por causa do nome e talvez participar de grupos tóxicos pela fama online. Vamos pensar um pouco sobre cada caso desses que eu mapeei durante esses últimos anos.

11.1 Vale a pena trocar de emprego para trabalhar com uma tecnologia em destaque?

Em um determinado momento da minha vida eu trabalhava com PHP e jQuery. Saí dessa stack (essa lista de tecnologias) para trabalhar com Angular e 100% no front-end. Antes eu trabalhava como fullstack. Essa não foi uma troca ruim. O mercado de

desenvolvimento de software estava mudando e o front-end estava ganhando uma certa moral que antes não tinha.

Antes de 2010, front-end era a pessoa que recortava layouts e somente isso. O máximo desafio de programação que essa pessoa tinha (na maioria das empresas) era validar formulários e isso me deixava bem longe dessa profissão. Não tinha interesse algum em recortar layout para a web. Porém, o mercado mudou e com o advento das *single page applications* (SPAs), aplicações em JavaScript rodando no navegador com muito mais complexidade do que antes, a profissão de pessoa engenheira front-end ganhou espaço.

Eu estava gostando do caminho que estava se criando e por isso foquei em me especializar nessa área. Foi uma troca muito feliz. Aproveitei o momento do mercado para me posicionar, mas bem mais para a frente, ali no final de 2014, como front-end. Essa troca era para que eu trabalhasse mais próximo do usuário final, focando em acessibilidade e performance de aplicações web, algo de que eu sempre gostei. Assim, minha jornada front-end começou em 2015.

Naquele momento estava trabalhando com Angular.js, um framework JavaScript que foi muito utilizado em diversas aplicações web complexas e me trouxe muitas experiências boas, assim como um excelente emprego. Porém, a comunidade de tecnologia já estava começando a falar mal de Angular.js e eu, como qualquer iniciante, passei a sentir um certo medo de continuar trabalhando com essa tecnologia e ficar defasado. Apareceu uma excelente oportunidade em uma grande empresa onde eu iria trabalhar com React. Na ansiedade de começar a trabalhar com a tecnologia de que todo mundo falava bem, eu logo aceitei a proposta.

O resumo da história foi que eu trabalhei durante 9 meses de volta com jQuery, algo que eu considerava um retrocesso em minha carreira profissional. O projeto que me havia sido prometido na realidade ainda estava no mundo das ideias e não tinha prazo nem para começar. Quando descobri que isso, acabei caindo na realidade:

eu havia trocado um emprego excelente, com ótimas perspectivas de crescimento e benefícios melhores, por uma promessa.

Muitas vezes vamos errar dessa forma. Acreditamos que o desafio técnico está na nova tecnologia de mercado ou em outra empresa maior, mas precisamos pensar melhor antes da troca de emprego, se esse for o motivo, pois ele é, basicamente, uma vaidade.

Claro que existem empresas cujas tecnologias estão muito defasadas e você nunca terá oportunidade de mudar algo ali dentro, porém, antes de tomar sua decisão, seria bom perguntar para a sua liderança se o futuro trará a oportunidade de modernizar as aplicações. Na maioria das vezes é só esperar um pouco e os sistemas serão modernizados.

Hoje, aquele produto em que eu trabalhava com Angular.js foi totalmente reescrito e esse seria um trabalho muito legal de se fazer, mas a minha falta de paciência e vontade de seguir o grupo acabou me colocando em um emprego onde eu só passei estresse e ainda trabalhei com tecnologia defasada. Atualmente, em 2022, o site onde fui trabalhar naquela troca continua com a mesma stack defasada e sem prazos para melhorias.

Lembre-se: a tecnologia não deve ser trocada somente pelo comportamento de manada, mas pelos benefícios que ela trará para o produto ou por conta dos problemas que temos com as ferramentas mais antigas. No caso desse segundo emprego, o resultado de continuar com a tecnologia defasada é trabalhar com um *codebase* muito grande, de difícil manutenção e com vários pontos de falha. Poderia ser resolvido com mais abstração e um trabalho maior de qualidade de código ou trocando de jQuery para outra ferramenta que força as boas práticas, mas não é interessante para a empresa mexer nesse pedaço agora.

O grande problema na troca nem é a tecnologia defasada, mas o tanto de estresse pelo que passei e os benefícios ruins que faziam com que meu salário nesta empresa fosse menor que o anterior,

mesmo tendo subido na carteira. Em São Paulo, não se almoçava o mês inteiro com um VR de 220 reais em 2017, então parte do meu salário ia para o almoço. Fora o desconto do convênio e outras questões que limitavam o meu faturamento líquido.

Trabalho é algo muito maior do que trabalhar com tecnologias da moda. O ambiente e salário deveriam ter contado mais do que isso para mim.

11.2 O que vale mais salário ou título?

Na nossa área temos a carreira em Y, onde podemos seguir por uma progressão técnica ou pelo caminho da gestão de projetos ou pessoas. Mas tem algo que é comum independente do caminho escolhido, os **níveis**.

Quando começamos na área, normalmente estamos como juniores ou em estágio, depois viramos pessoas nível pleno e, em seguida, seniores. Durante o nível intermediário (pleno) ou avançado (sênior) podemos escolher seguir em liderança técnica, liderança de projetos ou de pessoas. Dali para a frente, teremos cargos como: staff, sênior staff, principal e *distinguished* para níveis técnicos; gerente, sênior gerente, diretoria e vice-presidência para cargos de gestão.

A nossa vaidade ataca quando acontece o seguinte cenário: você é uma pessoa nível pleno e recebe uma proposta de nível sênior. Até aqui, tudo bem, é parte da sua progressão de carreira. O perigo está quando nós só enxergamos o título e não o contexto geral.

Voltando para o meu contexto, em determinado momento da minha carreira recebi duas propostas; uma para trabalhar como sênior front-end e outra para atuar como sênior staff. Visto o nome sênior staff, qualquer pessoa da área logo se sente bem, pois recebemos uma avaliação extremamente positiva. A empresa que me avaliou como sênior front-end me rebaixou, pois eu havia prestado o

processo seletivo **para staff**. Agora, vamos colocar na ponta do lápis (com valores fictícios para não expor as empresas), a empresa 1, que me avaliou como sênior, me ofereceu um salário acima de 20 mil, a empresa 2, que me avaliou como senior staff, me ofereceu o salário de 19 mil. O salário em si tem pouca diferença, os benefícios das duas empresas eram extremamente iguais, a grande diferença aqui era que, na empresa 1, eu teria mais 4 cargos para subir: staff, sênior staff, principal e distinguished; na empresa 2, eu teria somente mais 2 cargos.

Já vi muita gente trocando de emprego para trabalhar como sênior, e eu, como pleno, recebia um salário maior, tinha mais desafios técnicos, maior autonomia e um ambiente de trabalho mais seguro. O cargo não é nada. Algumas pessoas afirmam que às vezes vale a pena trocar de emprego para receber o título na carteira e depois ir para um emprego melhor, mas não é um cargo no seu LinkedIn que vai garantir que outra empresa avalie você no mesmo nível. Muitas vezes uma empresa nos avalia diferente do que está na nossa carteira. Não é isso que é importante no mercado de trabalho. O importante é o seu desempenho no processo seletivo, é isso que garante seu cargo. O grande risco aqui está em trocar de emprego, trocar o certo pelo duvidoso, somente por causa de status.

Quando fizer um processo seletivo e receber uma proposta melhor, mas com o título abaixo do que está atualmente, pense um pouco se título vale mais do que sua saúde mental ou o seu futuro profissional.

11.3 Trabalhar em uma empresa grande para ter o nome no currículo vale a pena?

No nosso mercado, acredito que, como qualquer outro, existem empresas que são as queridinhas das pessoas desenvolvedoras. Seja por uso de tecnologias modernas, por suas práticas de engenharia extremamente eficientes ou mesmo por um *employer branding* muito bom, o marketing atrás da imagem da empresa para atrair recursos.

Em um determinado momento da minha vida profissional, coloquei uma empresa como um sonho de ambiente de trabalho. Isso porque as pessoas daquele lugar estavam em todos os eventos de tecnologia possíveis palestrando sobre como suas práticas de engenharia são eficientes. Eu imaginava o quão desafiador seria estar neste ambiente, onde milhares de clientes estavam conectados por minuto e o software escalava muito rápido para aproveitar o mercado aquecido para aquele modelo de negócios.

Foi muito legal fazer o processo seletivo dessa empresa, pois percebi a desorganização daquele local. Eu, como desenvolvedor front-end, recebi um teste de back-end, assim como um amigo back-end recebeu o desafio de mobile. Quando questionados, responderam que por lá não tem distinção do que cada pessoa engenheira faz, são todos desenvolvedores de software e você tem que se virar para aprender aquilo em que vai atuar quando entrar lá. Por mim, ok, não sou contra a cultura fullstack, que é quando não temos especialização e trabalhamos de ponta a ponta nas entregas de software. Porém, o processo seletivo poderia levar em consideração que a minha maneira de trabalhar desde 2015 era como front-end, não como back-end, e do meu amigo era como back-end, não como mobile.

O teste acabou ficando muito mais estressante do que deveria ter sido. Este foi só o começo. Depois descobrimos que o ambiente de trabalho era extremamente tóxico, cheio de demandas *topdown*, onde você não tem autonomia nenhuma, e o único benefício real de se trabalhar naquele lugar era o salário muito alto que pagavam para segurar as pessoas lá.

No meu caso, eu nem queria um nome específico no meu currículo, como Google ou Microsoft, mas sim trabalhar em um lugar que me havia me conquistado pelo nível técnico - e mesmo assim isso seria uma furada. O mesmo acontece com pessoas que vão para as big techs acreditando que terão o emprego dos sonhos, mas acabam caindo em projetos nada atrativos e equipes não muito legais. Isso acontece porque empresas grandes não têm um padrão real de como todas as equipes trabalham, você pode cair naquela equipe legal ou projeto fantástico, mas pode acontecer o contrário.

Agora vem a grande questão, ter um nome grande no currículo faz diferença no nosso futuro? Claro que faz. Algumas empresas costumam procurar por pessoas que trabalham em seus concorrentes diretos. A dúvida que fica em nossas mentes é se vale a pena arriscar entrar em um emprego que pode não ser muito legal se conseguimos destaque para arrumar boas oportunidades de outras formas também, como ter um blog técnico, palestrar ou participar ativamente das comunidades.

11.4 Participar de grupos tóxicos porque eles estão lotados é bom?

Chegando até aqui você sabe que eu adoro as comunidades de tecnologia. Porém, existem grupos e grupos. Existem comunidades de 500 pessoas com um clima muito legal, rede de apoio e união de verdade. Mas também temos grupos de mais de 100 mil pessoas cujo ambiente é tão tóxico que parece um bueiro de usina nuclear.

O grande problema aqui é que, quando queremos ganhar visibilidade na comunidade, 500 pessoas podem não dar a quantidade de "likes" que queremos. Estar em um grupo com

milhares de pessoas vai dar a sensação de fama que muita gente quer sentir. É claro que eu já passei por isso. Em um determinado momento do meu blog eu comecei a compartilhar as postagens somente naquele grupo maior de JavaScript, onde tinha pessoas que falavam coisas absurdas sobre minorias, eram bem machistas e, em sua maioria, nenhum se identificava comigo. Mas esse grupo, que era ruim para mim, fazia os gráficos de acessos do meu blog crescerem muito. Isso alimenta o ego de qualquer pessoa.

Mas, será que realmente valeu a pena estar naquele espaço? Olhando em retrospecto, fora os acessos momentâneos, quando eu compartilhava os links no fórum, durante o resto das semanas eu só vivia situações que me faziam mal por estar dentro daquela comunidade. A internet se tornou parte das nossas vidas e em diversos momentos eu passava estresse do começo até o final do dia por causa dos comentários daquelas pessoas e do que incentivavam. Hoje eu percebo que nada é melhor do que ter paz. Muitas vezes eu deixei de estudar, de me divertir ou curtir meu relacionamento por causa do estresse pelo qual estava passando. Isso simplesmente não vale a pena.

Conclusão

É muito difícil não tomar decisões ruins baseadas no nosso ego uma vez que entender o que é vaidade e o que é razão no que fazemos vem com as experiências ruins e com muita terapia. O mais importante aqui é sempre analisar suas atitudes e como elas estão refletindo em sua vida agora e como vão atingir você no futuro.

Trocar de emprego por causa de tecnologia? Talvez, mas analise antes. Salário, ambiente de trabalho legal, bons benefícios ou título? Bom, você já sabe o que eu penso sobre isso, títulos não valem quase nada na nossa área. Big techs ou outras empresas legais? Tanto faz, desde que você entre em um emprego que lhe faça bem. Comunidades tóxicas? Fuja delas.

Espero que este capítulo e o compartilhamento das minhas histórias de vida ajudem você a evitar alguns problemas de carreira em programação.

Capítulo 12
Preconceito com tecnologias

O ser humano é cheio de preconceitos. Não tem como fugirmos, pois é parte da nossa criação em grupos que podem pensar muitas e muitas coisas. Eu e você temos experiências de vidas e crenças que moldam nossa maneira de se comportar e até mesmo o modo como vivemos em sociedade e parte disso veio do meio em que crescemos.

Um preconceito nada mais é que uma opinião baseada em algo sem um exame crítico disso. E essa opinião pode até mesmo levar a atitudes hostis, agressividade ou intolerância com alguma coisa ou com pessoas. Assim como acontece com o racismo, intolerância religiosa, misoginia, LGBTfobia e muitos outros grupos que são alvo de ataques por simples preconceito.

No capítulo 10, **Por que existem cursos de programação exclusivos para LGBTs e outras minorias?**, comentei sobre como a comunidade de programação pode ser excludente com alguns grupos sociais. Neste capítulo, quero refletir com você sobre como nós podemos acabar perdendo boas oportunidades de experiências legais por conta do preconceito com tecnologias, que por alguns é conhecido como **preconceito tecnológico**.

Existem diversos momentos em nossas vidas em que aprendemos os preconceitos que, depois, precisamos desconstruir. Como pessoas engenheiras de software, o bom desenvolvimento de um produto é (quase sempre) independente de tecnologia e precisamos escolher a ferramenta certa para o problema que estamos tentando resolver. Para uma boa escolha, não podemos ter ideias erradas e sem fundamentos sobre possíveis soluções. Vamos caminhar um pouco sobre alguns dos momentos em que aprendemos vieses e pensar como eles impactam em nossa carreira.

12.1 O que aprendi na faculdade é a verdade universal

A maioria dos cursos relacionados a Ciência/Engenharia da Computação possui uma matéria em sua grade chamada Linguagens de Programação, onde você aprende de fato alguma linguagem. Porém você pode aprender em outra cadeira também, como em Algoritmos e Estruturas de Dados ou Introdução a Lógica de Programação. É neste momento que a grande maioria das pessoas é apresentada a sua **primeira linguagem de programação**. A partir dali, talvez esta se torne a sua linguagem favorita, dependendo de como lhe foi apresentada.

Durante a graduação, em um momento um professor comentou:

> Vamos aprender Java, pois Java é a linguagem do futuro!

Eu ouvi este comentário em 2020, depois de alguns anos trabalhando com tecnologia e com conhecimento suficiente para revisar essa frase com criticidade suficiente para entender que esse papo de linguagem do futuro, a única linguagem que você precisa aprender, é furada.

Java é uma linguagem muito versátil, universal e extremamente requisitada nas empresas de tecnologia e produtos, mas não caia na ilusão de que será a única ferramenta que você deve aprender e utilizar pelo resto da vida para qualquer cenário.

Uma pessoa que só sabe utilizar uma marreta vai tentar colocar um parafuso na base da marretada. Isso é correto?

O papel da faculdade é apresentar conceitos que, normalmente, você não vai encontrar no mercado de trabalho. Pouca gente vai ter contato com tudo o que é ensinado por lá. Isso é legal e muito útil,

mas você não pode deixar se levar pelo comentário de alguns professores que podem nos enviesar para o resto da vida.

Professores são seres humanos que possuem seus preconceitos e vão reproduzir em algum momento.

Lendo um livro de Programação Funcional, recentemente, um professor de uma grande universidade brasileira comenta que JavaScript é uma linguagem utilizada unicamente para a manipulação de formulários. Isso não somente está errado, quanto vai enviesar diversos alunos e alunas de seu curso, levando-os a acreditar que não se usa essa linguagem para nada mais e o mercado é bem diferente dessa visão.

Claro que, às vezes, um professor ou professora pode fazer uma piada com uma tecnologia por conta de ser uma convenção da comunidade, como quando alguns dizem que Java é lento. Quando você ouvir algo assim e a pessoa não complementar dizendo que é só uma brincadeira baseada em um comentário vazio que muita gente compartilha, é necessário pensar:

- Por que Java é lento?
- O que significa ser lento para uma linguagem de programação?
- Baseado em quais estudos conseguimos identificar que uma linguagem é lenta?
- A performance de uma linguagem impacta onde?

Precisamos revisar o comentário com mais criticidade, independente de ele vir de uma pessoa que está em uma posição alta no compartilhamento de conhecimento, como professores e professoras. Sempre busque informação para validar o que está aprendendo.

12.2 O que está escrito no livro deve ser

seguido à risca

Livros são excelentes fontes de conhecimento. Antes de entrar em uma faculdade, eles foram minhas bases em Ciência da Computação que eu não encontrava nos cursos e artigos que estava consumindo.

O problema não está no conteúdo do livro - OK, em alguns momentos, como no exemplo citado acima, está. O que pega mal no desenvolvimento de software é seguir o livro à risca, esquecendo que cenários mudam, tecnologias evoluem e os problemas que enfrentamos no dia a dia podem ser únicos/diferentes do que está no livro.

Existe um termo para quando queremos implementar algo exatamente como nos livros, chama-se *"by the book"*. Em diversos momentos na comunidade podemos encontrar discussões porque alguém está tentando seguir uma convenção que aprendeu na última leitura, deixando de lado o cenário controlado que foi planejado para a escrita do livro.

Temos conceitos que não tem como não seguir *by the book*, por exemplo, quando estamos implementando um algoritmo clássico. Se você mudar o comportamento do algoritmo, você não está escrevendo aquele que aprendeu, mas uma mutação dele. Mas, no dia a dia como pessoas desenvolvedoras de software, podemos enfrentar momentos onde alo não se aplica.

Vamos pensar no *Clean Code* (Código Limpo, de Robert Martin), um livro famoso na área de programação... Em alguns momentos a pessoa abstrai tanto um código pensando na não repetição que aquilo se torna uma caixa preta. Quando você lê o código, não consegue relacionar nada do que está acontecendo por ali com as regras de negócio que estão sendo aplicadas. O debug desse código não será tão legal. Isso acontece por causa de um outro conceito chamado **over abstraction**.

Uma vez, um desenvolvedor mais experiente comentou em um *pull request* que abri: "uma regra de negócio é uma regra de negócio, não é uma abstração de engenharia de software". Essa frase acabou me acompanhando sempre que estou pensando na reutilização de código. Nem tudo o que escrevemos é para o software, a maioria das linhas de código, na realidade, são para outra pessoa desenvolvedora ler e é pensando nisso que o *Clean Code* foi escrito. Em alguns momentos, é tudo bem repetir um código para privilegiar o entendimento do *code base* acima da abstração desmedida.

Isso também acontece quando estamos seguindo o scrum. Algumas pessoas reclamam de que, em sua empresa, o scrum não é seguido à risca, mas será que isso é necessário? O modelo que sua empresa está utilizando pode ser o melhor para o tamanho da equipe que você trabalha, para os requisitos legais que ela precisa atender ou qualquer burocracia que exista em cima do seu modelo de negócios.

O mundo é maior do que a quantidade de exemplos que conseguimos demonstrar em livros, artigos ou palestras, você precisa utilizar o que ensinamos como base para seguir, não como regras que não podem mudar dependendo do contexto em que você está.

12.3 Disseram-me que X é ruim

Algo em que já vi muita gente iniciante se dando mal foi pelo fato de seguirem muita gente em redes sociais que falam coisas aleatórias sobre tecnologias e acabarem copiando essa fala. Lembra que nossos preconceitos, normalmente, vêm do contexto onde estamos? Quanto mais gente reproduz um preconceito, maior a chance de isso chegar a nós e acabarmos copiando o comportamento.

Por diversas vezes você verá pessoas comentando em redes sociais que Java é lento, que JavaScript é ruim por não ter tipos, que

Python só serve para trabalhar com projetos acadêmicos, que Windows não presta para pessoas programadoras, que você precisa ter um blog... São centenas de frases que são repetidas por pessoas aleatórias **todos os anos**.

É sério... Desde que estou na área de programação e a partir do primeiro dia em que me conectei com as comunidades de tecnologia, já comecei a receber essas premissas e, o pior, também assumi várias delas como verdadeiras.

Algumas coisas são verdadeiras dependendo do contexto em que você está. Um dos casos mais aleatórios nesse ciclo de repetição anual de discussões é o do sistema operacional. Pense comigo: se o Windows funciona para o seu trabalho, você tem produtividade igual em outros sistemas, por que ele é ruim?

Em um determinado momento da minha carreira, eu atuei no back-end com PHP, depois com Python e, posteriormente, com C#. Algo que se repetiu nas experiências que tive com essas tecnologias foi o fato de as pessoas que trabalhavam comigo falarem mal de Java. É engraçado, mas perigoso. Como um bom iniciante, eu assumi que o que as pessoas sêniores estavam falando era a verdade absoluta que eu deveria seguir para minha vida.

O que eu não esperava é que trabalharia, depois disso, em várias empresas que utilizam Java em suas aplicações de maior impacto/importância. Um certo dia, perguntei para a pessoa em maior nível técnico dentro da empresa por que utilizávamos uma linguagem "lenta". Para mim, estava certo que Java é uma tecnologia lenta, o meu erro era em não saber o que de fato é uma tecnologia lenta e muito menos ter feito testes para verificar se isso é real **em Java**. Essa pessoa me apresentou diversos motivos pelos quais essa tecnologia foi adotada na empresa, e uma garantia é que ela não é lenta para os casos de uso daquele produto.

Perceba: em diversos momentos as pessoas vão falar coisas que parecem ser muito reais, parece ser algo realmente com

fundamentos. O ponto que você deve sempre ter em mente é se esses fundamentos existem mesmo. Existem testes que provam o que essa pessoa diz? Nós trabalhamos com tecnologia, não com previsão do futuro, essas pessoas precisam trazer provas do que estão falando, pois faz parte do método científico, que é de onde vêm nosso conhecimento e nossas convenções.

Conclusão

Tudo depende. Quanto mais experiência você tiver, e experiência não é tempo de carreira, mas desafios que enfrentamos e coisas diferentes que fazemos, mais vezes você vai trocar a verdade absoluta pelo termo **depende**.

A maioria das decisões em Engenharia de Software e Ciência da Computação *depende*. O contexto em que um desafio se encontra pode ditar qual a tecnologia que vamos utilizar, como linguagem de programação, framework, ambientes, sistemas operacionais e muito mais. Existem realmente muitas opções no mercado de programação e é por isso que tudo depende.

A tecnologia é ruim quando foi escolhida para resolver um problema para o qual ela não se propõe a resolver. Um exemplo que está dentro do meu contexto atual utilizando a tecnologia Node.js é o processamento intenso de arquivos, como fotos. Até o momento da escrita deste livro, utilizar as APIs do Node.js para processar imagens não é a melhor decisão que podemos tomar. O cenário ideal seria utilizar o Node em seu domínio, que é o tratamento de requisições web como webservice, e implementar uma camada em C, Rust ou Golang, que são tecnologias excelentes para processamento intenso.

Isso significa que Node é ruim? Não. Significa que *nenhuma tecnologia é uma bala de prata*, ditado utilizado muitas vezes na nossa área quando pessoas desenvolvedoras querem utilizar a mesma ferramenta para resolver qualquer problema possível.

Quando chegar um desafio que você acredita que determinada tecnologia é a melhor para resolver, utilize o método científico e observe o problema, formule perguntas que precisam ser resolvidas para atender a essa demanda, crie hipóteses de resolução, experimente muito, depois analise o resultado das experimentações para, enfim, tirar uma conclusão. É claro que outras pessoas já podem ter feito isso e podemos simplesmente seguir o que dizem, porém, você precisa trabalhar com criticidade para verificar se os fundamentos do que essa pessoa diz são reais ou são baseados em conclusões e preconceitos pessoais.

Capítulo 13
Conclusão

Considerações antes de você partir.

Este livro é uma coletânea dos artigos que escrevi para os meus mentorados e mentoradas tentando lhes tranquilizar para uma carreira mais calma. Espero de coração que esses conselhos tenham servido para você tanto quanto ajudou as pessoas que eu já instruí pessoalmente.

A partir de agora é com você! A sua jornada pode ser mais difícil ou mais fácil, não julgue quem está do lado com os olhos de quem está por fora. Evite o julgamento, este é o pior comportamento da nossa espécie. Pessoas no começo de carreira podem ter desafios iguais ou mais "fáceis" que os seus, ajude sem se importar com isso.

Espero que você encontre um emprego legal, que lhe faça bem e não leve você ao esgotamento. Se quiser empreender, torço para que seja um grande sucesso. Se quiser viver de freelance, que nunca lhe faltem clientes. Mas, busque cuidar da sua saúde mental e física acima de tudo. Isso não só vai ajudar você a viver melhor, mas também a produzir bem, estudar e não se acabar por causa do estresse e pressão da rotina de trabalho.

A área de tecnologia é estressante, mas conseguimos escapar disso quando abrimos o jogo. Converse sempre com sua liderança e não deixe o estresse lhe impactar tanto. Se sentiu mal, corre pedir ajuda. Não recebeu acolhimento? Vem atrás da comunidade de programação, nós estamos aqui para ajudar.

Compartilhe esse conhecimento com outras pessoas. Se o conhecimento ficar preso com você ou comigo, de nada ele vale. O conhecimento existe para ser compartilhado e gerar impacto social.

Boa sorte na sua caminhada e conte comigo nas redes sociais.

- Twitter: @1ilhas (https://twitter.com/1ilhas)
- YouTube: O Universo da Programação (http://bit.ly/yt-universo-da-programacao)
- Twitch: @1ilhas (https://twitch.tv/1ilhas)